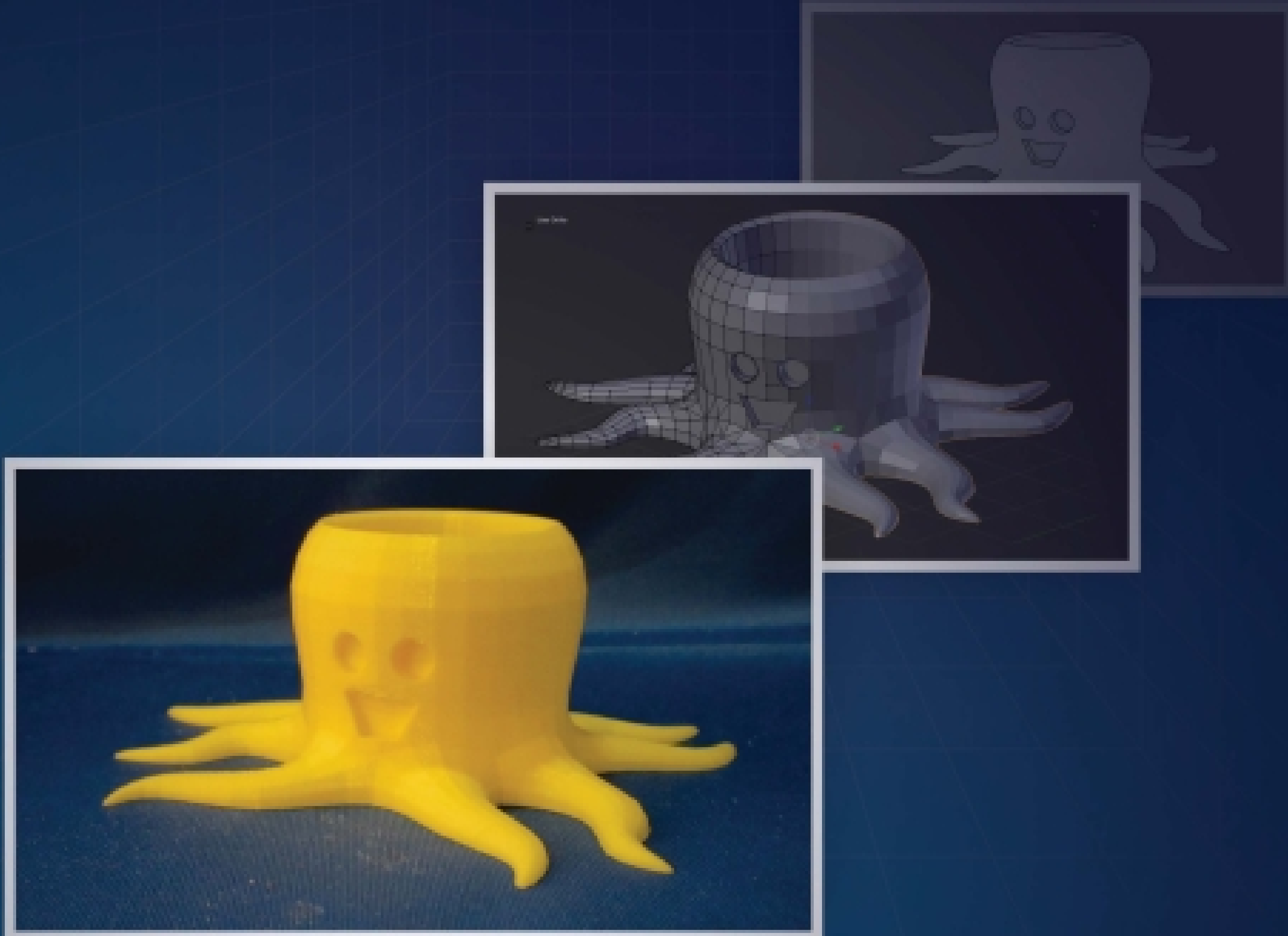


Community Experience Distilled

3D Printing Designs: Octopus Pencil Holder

Learn to design and 3D print organic and functional designs using Blender

Joe Larson

[PACKT] PUBLISHING open source* community experience distilled

Table of Contents

3D Printing Designs: Octopus Pencil Holder

3D Printing Designs: Octopus Pencil Holder

Copyright © 2016 Packt Publishing

All rights reserved. No part of this book may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, without the prior written permission of the publisher, except in the case of brief quotations embedded in critical articles or reviews.

Every effort has been made in the preparation of this book to ensure the accuracy of the information presented. However, the information contained in this book is sold without warranty, either express or implied. Neither the author, nor Packt Publishing, and its dealers and distributors will be held liable for any damages caused or alleged to be caused directly or indirectly by this book.

Packt Publishing has endeavored to provide trademark information about all of the companies and products mentioned in this book by the appropriate use of capitals. However, Packt Publishing cannot guarantee the accuracy of this information.

First published: February 2016

Production reference: 2260416

Published by Packt Publishing Ltd.

Livery Place

35 Livery Street

Birmingham B3 2PB, UK.

ISBN 978-1-78588-517-4

www.packtpub.com

Credits

Author

Joe Larson

Reviewer

Marcus Ritland

Commissioning Editor

Edward Gordon

Acquisition Editor

Vinay Argekar

Content Development Editor

Shweta Pant

Technical Editor

Vishal Mewada

Copy Editor

Stuti Srivastava

Project Coordinator

Kinjal Bari

Proofreader

Safis Editing

Indexer

Priya Sane

Graphics

Kirk D'Penha

Production Coordinator

Shantanu N. Zagade

Cover Work

Shantanu N. Zagade

About the Author

Joe Larson is one part artist, one part mathematician, one part teacher, and one part technologist. It all started in his youth when he worked on a Commodore 64, doing BASIC programming and low-resolution digital art. As technology progressed, so did Joe's dabbling, eventually taking him to 3D modeling while in high school and college, and he temporarily pursued a degree in computer animation. He abandoned this field for the much more sensible goal of becoming a math teacher, which he accomplished when he taught 7th grade math in Colorado. He now works as an application programmer.

When Joe first heard about 3D printing, it took root to his mind, and he went back to dust off his 3D modeling skills. In 2012, he won a Makerbot Replicator 3D printer in the Tinkercad/Makerbot Chess challenge with a chess set that assembles into a robot. Since then, his designs on Thingiverse have been featured on Thingiverse, Gizmodo, Shapeways, Makezine, and other places. He currently maintains the blog http://joesmakerbot.blogspot.in/, documenting his adventures.

About the Reviewer

Marcus Ritland is a designer and 3D printing consultant in his small business, Denali 3D Design. Since 2008, he has been providing 3D modeling and 3D printing services as well as moderating the SketchUcation 3D printing forum (http://sketchucation.com/).

He has volunteered at a local makerspace, teaching SketchUp classes and leading 3D printing meetups. As an author of 3D Printing with SketchUp, he is currently on a quest to eliminate design-for-3D printing illiteracy.

www.PacktPub.com

eBooks, discount offers, and more

Did you know that Packt offers eBook versions of every book published, with PDF and ePub files available? You can upgrade to the eBook version at www.PacktPub.com and as a print book customer, you are entitled to a discount on the eBook copy. Get in touch with us at <customercare@packtpub.com> for more details.

At www.PacktPub.com, you can also read a collection of free technical articles, sign up for a range of free newsletters and receive exclusive discounts and offers on Packt books and eBooks.

https://www2.packtpub.com/books/subscription/packtlib

Do you need instant solutions to your IT questions? PacktLib is Packt's online digital book library. Here, you can search, access, and read Packt's entire library of books.

Why subscribe?

- Fully searchable across every book published by Packt
- Copy and paste, print, and bookmark content
- On demand and accessible via a web browser

Preface

3D printers have arrived! Complex and beautiful objects are available at the touch of a button in our schools, libraries, or even our homes. Whether or not you have a 3D printer, learning how they work and how to design for them is the best way to be a part of this new industrial age. And the best part is it doesn't cost a penny.

This book will teach you the things you need to know about 3D printers. Then, you will use the robust and free software, Blender, to follow step-by-step instructions through a planned project. This book is part of a series of projects that will help you acquire the tools, techniques, and skills you need to make your own projects that you can print by yourself on a 3D printer near you and share with others online to print around the world.

What this book covers

Chapter 1, *3D Printing Basics*, will help you understand 3D printing basics, types of 3D printing, and how FFF printers work.

Chapter 2, *Beginning Blender*, will introduce Blender, how to set it up, and some basic and mid-level functionality. Knowing the content of this chapter will get you over Blender's infamous learning curve and provide the basic knowledge and reference necessary for following along with future projects.

Chapter 3, *The Octopus Pencil Holder*, the project in this book, and octopus pencil holder, will involve simple selection techniques unique to Edit mode, modification commands in Edit mode, and applying modifiers to soften and combine shapes. This technique alone can be used to make an unlimited number of cool things once mastered.

What you need for this book

Blender has these minimum system requirements:

- A 32-bit dual-core 2-GHz CPU with SSE2 support
- 2 GB of RAM
- A 24-bit 1280×768 display
- A mouse or trackpad
- OpenGL 2.1 compatible graphics with 512 MB of RAM

Who this book is for

This book is for anyone with an interest in 3D printing and some basic computer skills. Whether you own a 3D printer or not, you can design for them. You will need Blender, a free 3D tool, and this book. With a little creativity, one day you'll hold in your hands something designed with a computer.

Conventions

In this book, you will find a number of text styles that distinguish between different kinds of information. Here are some examples of these styles and an explanation of their meaning.

Code words in text, database table names, folder names, filenames, file extensions, pathnames, dummy URLs, user input, and Twitter handles are shown as follows: Windows and unsure, just choose the `MSI` package option.

New terms and **important words** are shown in bold. Words that you see on the screen, for example, in menus or dialog boxes, appear in the text like this: Locate the **Blender** download button on the main page for the latest version of Blender and click on it.

> **NOTE**
>
> Warnings or important notes appear in a box like this.

> **TIP**
>
> Tips and tricks appear like this.

Reader feedback

Feedback from our readers is always welcome. Let us know what you think about this book—what you liked or disliked. Reader feedback is important for us as it helps us develop titles that you will really get the most out of.

To send us general feedback, simply e-mail <feedback@packtpub.com>, and mention the book's title in the subject of your message.

If there is a topic that you have expertise in and you are interested in either writing or contributing to a book, see our author guide at www.packtpub.com/authors.

Customer support

Now that you are the proud owner of a Packt book, we have a number of things to help you to get the most from your purchase.

Downloading the color images of this book

We also provide you with a PDF file that has color images of the screenshots/diagrams used in this book. The color images will help you better understand the changes in the output. You can download this file from https://www.packtpub.com/sites/default/files/downloads/3DPrintingDesigns_OctopusPencilHolder_ColorImages.pdf.

Errata

Although we have taken every care to ensure the accuracy of our content, mistakes do happen. If you find a mistake in one of our books—maybe a mistake in the text or the code—we would be grateful if you could report this to us. By doing so, you can save other readers from frustration and help us improve subsequent versions of this book. If you find any errata, please report them by visiting http://www.packtpub.com/submit-errata, selecting your book, clicking on the **Errata Submission Form** link, and entering the details of your errata. Once your errata are verified, your submission will be accepted and the errata will be uploaded to our website or added to any list of existing errata under the Errata section of that title.

To view the previously submitted errata, go to https://www.packtpub.com/books/content/support and enter the name of the book in the search field. The required information will appear under the **Errata** section.

Piracy

Piracy of copyrighted material on the Internet is an ongoing problem across all media. At Packt, we take the protection of our copyright and licenses very seriously. If you come across any illegal copies of our works in any form on the Internet, please provide us with the location address or website name immediately so that we can pursue a remedy.

Please contact us at <`copyright@packtpub.com`> with a link to the suspected pirated material.

We appreciate your help in protecting our authors and our ability to bring you valuable content.

Questions

If you have a problem with any aspect of this book, you can contact us at <`questions@packtpub.com`>, and we will do our best to address the problem.

Chapter 1. 3D Printing Basics

As cool as 3D printing is, there is a lot of hype around it, which sometimes causes confusion. Before starting to design for 3D printing, it's best to know a little bit about 3D printing technologies.

3D printing is a limitless technology in the sense that there is no end to the things it can make. Still, that doesn't mean that it can make anything without limitations. 3D printing can make things that no other manufacturing method can, but it has rules that need to be followed to ensure success. There are different types of 3D printing as well, and each type comes with its benefits and drawbacks:

In this chapter, we'll discuss:

- What is 3D printing?
- What types of 3D printing are there?
- How do FFF printers work?
- The anatomy of an FFF print.
- Supportless 3D printing and YHT.
- Wall thickness and tolerances.

What is 3D printing?

3D printing is cool. It seems as if not a day passes without another mention of 3D printing online in the news and media. Everyone is getting excited about 3D printing. But when you look deeper, it seems as if everything is being 3D printed, and anything could be. Does 3D printing

something make it better? What exactly is 3D printing?

In many ways, 3D printers are just tools, the same as any that you'd find in a wood shop or garage. These tools make cool things, but not on their own, and just because something is made with, say, an electric drill press, that doesn't automatically make it better than something that isn't. It's the things that people, like you, are doing with these tools that make them cool.

I'm not saying that 3D printing isn't cool by itself. 3D printing lets you create things, test them, change their design, and try something new quickly until you get it right. It makes things of incredible complexity and, because it's additive manufacturing, generates comparatively little waste. The availability of cheaper and faster 3D printers means that there's a chance that there's a 3D printer near you.

What defines 3D printing?

There are many different types of 3D printers, but what makes them all similar is that they build solid shapes from layers of materials, starting with an empty build area and filling it with the print. This is called **additive manufacturing**, and it produces less waste than other techniques, such as starting with a base material that is cut away to make the thing.

3D printers also benefit from being computer-controlled machines, also known as **computerized numerical control** (**CNC**) machines, meaning they do what they do with minimal human interaction after the design work is done. They can make many identical copies of a thing one right after the other, and the design

can be shared online so that others can make their own copies.

While all 3D printing shares come common features, there are several distinct types of 3D printing that vary in how they produce the print. **Fused filament fabrication** (**FFF**), **powder bed**, or **light polymerization**, for example, all accomplish 3D printing in very different ways, and each with their own strengths and weaknesses. What works in powder bed 3D printing might not work with FFF 3D printing, and the part you get from light polymerization might not be suitable for the same usage as those made with the other techniques.

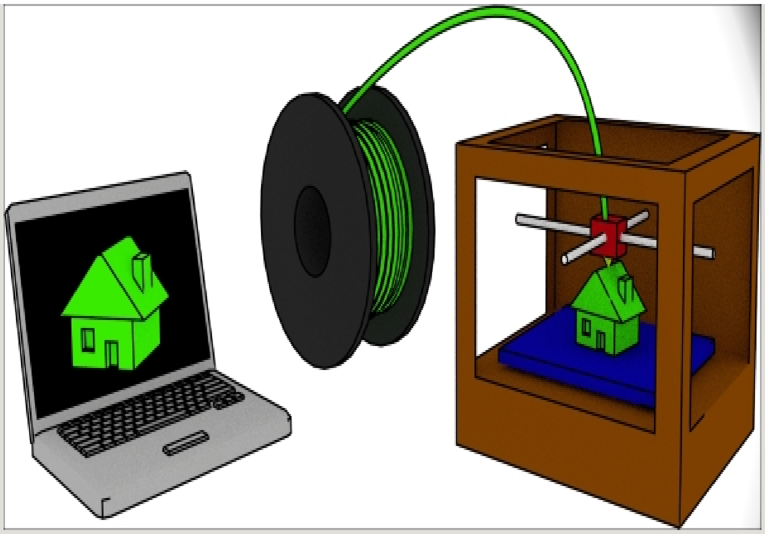
What to design for?

It is the best practice to always design towards the strengths and weaknesses of the medium you'll be using. The projects in this series of books will focus on designing for FFF 3D printers, because they're inexpensive and more readily available than the others, and the parts made with FFF 3D printers are suitable for a wide variety of functional uses. Also, many of the techniques for FFF design transfer to the other types of 3D printing. But because FFF 3D printers have limitations, there will be some things you need to know first.

How do FFF printers work?

There tends to be a lot of variation within the family of FFF 3D printers. Some have their mechanisms exposed to the environment so that they're easy to repair, while some are protected with fancy covers so that they look good. Some have one extruder, while some have two or more. Some have fancy interface screens, and some require you to use a computer to access even the most basic functions. Yet, for all their variations, there are many similarities that all FFF printers share which define their type. Being familiar with how FFF 3D printers work will help you guide yourself while designing for them.

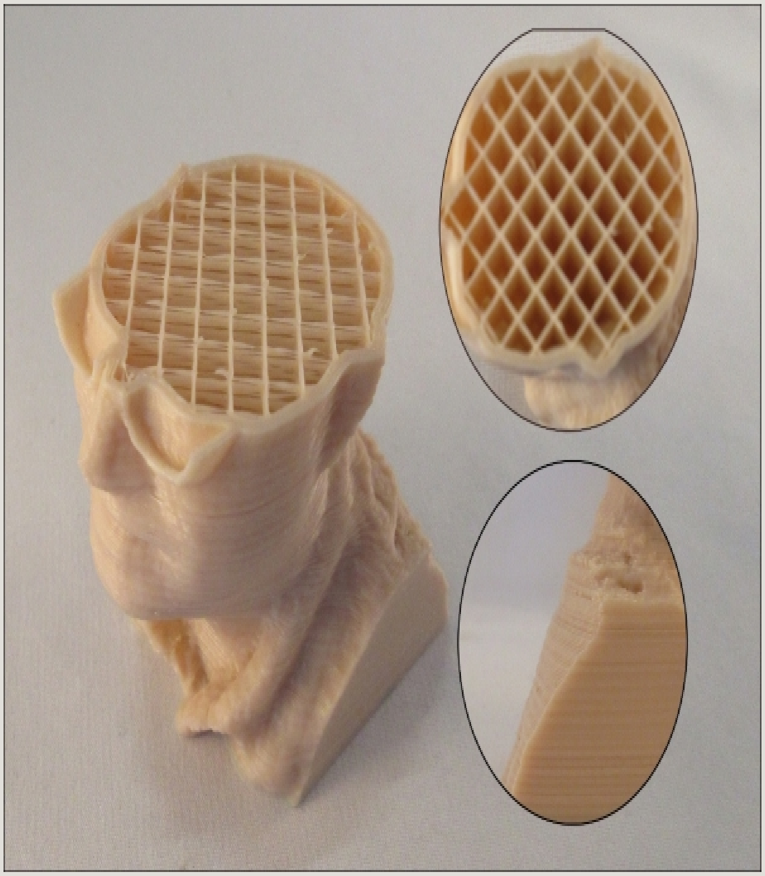
For FFF 3D printers, a computer takes a 3D model and translates it into commands that the printer can follow. The printer then takes a roll of plastic filament on a spool and uses a feeder mechanism to feed it into the hot end, where the plastic filament is melted and squirted out at a controlled rate onto the print bed, where the print is built up. The extruder head and print bed are moved relative to each other in 3 dimensions, using some sort of movement system in order to create the 3D model:



Drawing a print layer by layer like this takes, as might be expected, a little bit of time. The larger the object, the longer a print will take. FFF 3D printing isn't a fast process. But once the process is done, a new thing will have been created.

The anatomy of a print

Now that the mechanics of FFF 3D printing are clear, it's time to take a look at how a print is built. If an FFF print is stopped partway through, or observed during printing, the following can be seen:



The following are the different parts shown in the preceding image:

- **Layers**: FFF prints in layers, with each layer sitting on the one below it. Prints can be made with thicker layers so that they print faster or thinner layers so that they look better.

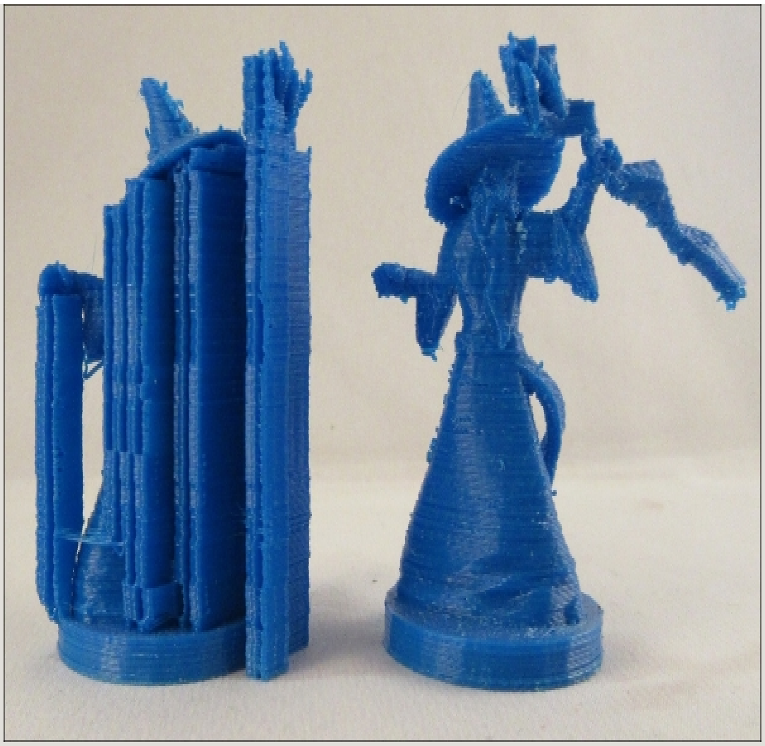
- **Outlines**: When starting a layer, the outline of that layer will usually be printed first. FFF prints often have two or more outlines so that the outside of the print is strong.
- **Infill**: once the outline is done, the rest of the layer is filled in. If an area of the print will not be seen from the outside when the print is done, a loose infill is used to save material and give layers above something to sit on. Top layers are filled in completely. Most FFF prints are largely hollow.

FFF design considerations

The basic limitations of FFF printers stem from the fact that most FFF 3D printers are developed by people who have very little accountability. To the people creating and manufacturing these printers, if the printer can print a thing most of the time, then that's probably good enough. In this way, FFF printers are more like garage tools than desktop machines. For those unfamiliar with FFF printers, there are some drawbacks that need to be taken into account.

Overhangs and supports

FFF 3D printers have to worry about overhang. Overhang is when a part of the design, when it prints, will not have anything between it and the build platform. To compensate for this, the 3D printer can build a lattice of support material up to the overhanging part. After the print, the support material will have to be removed. But since for most FFF 3D printers the support material is made of the same material as the object, it can rarely be removed without a trace that is sometimes difficult to clean up completely and can leave a mess on more complex prints:



Because of the troubles with supports, it's a good idea to design for supportless 3D printing.

Supportless 3D printing

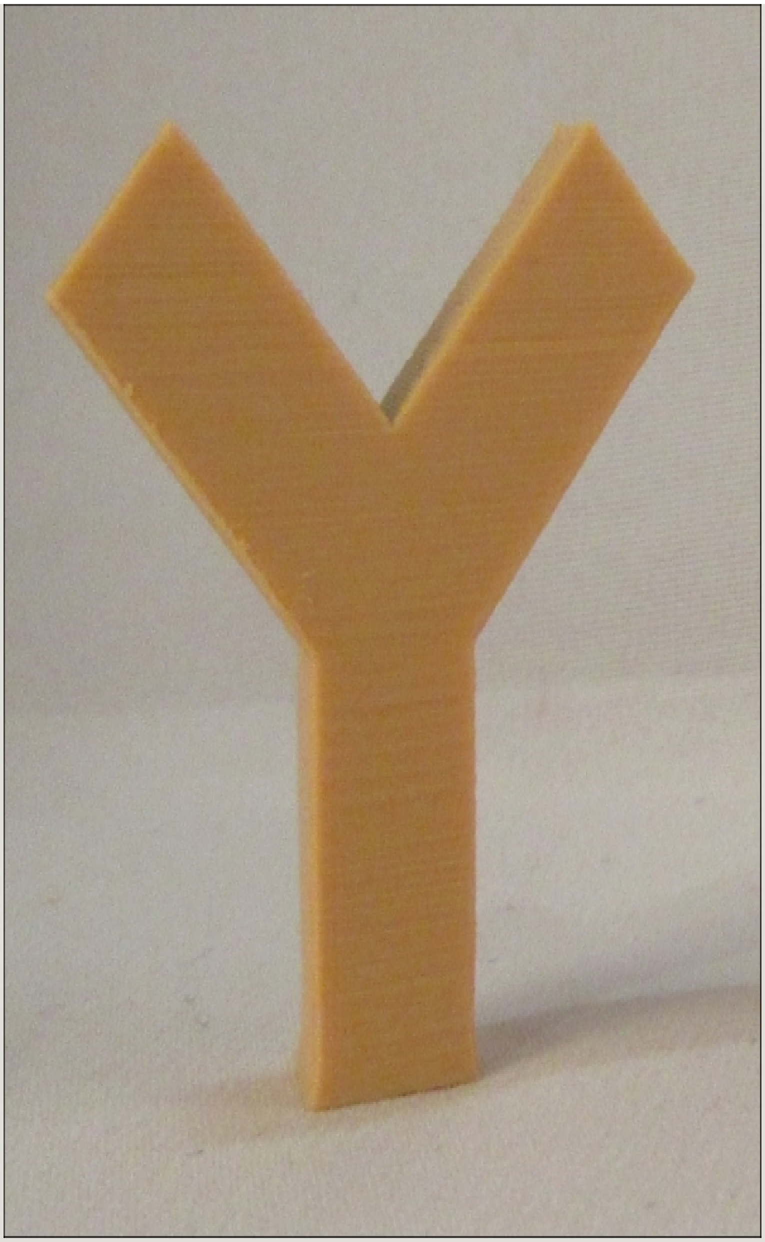
Think about building a snowman or sand castle. There's a lot that can be done with the medium of sand or snow, but try to get too fancy with the design and it will fall apart. As long as every part is sitting on top of something, chances are it will hold together. You could even slope gently outwards, as long as you don't push it too far.

It's the same with 3D prints. Because it prints in layers, each layer needs to have something to lay down on. If a design is made so that a part has nothing underneath it and is dangling in the air, then the printer will still extrude some plastic to try to print the part, but with nothing to print on, the plastic will just drool from the extruder until it gets wiped off on some other part, making an ugly mess and ruining the print.

As long as you put some thought into it, you can make designs that will succeed in most cases. There are a few rules that can help, and these rules can be illustrated with the letters Y, H, and T.

Y – gentle overhangs

Think about 3D printing a capital letter Y, standing up on the build platform—something like this:



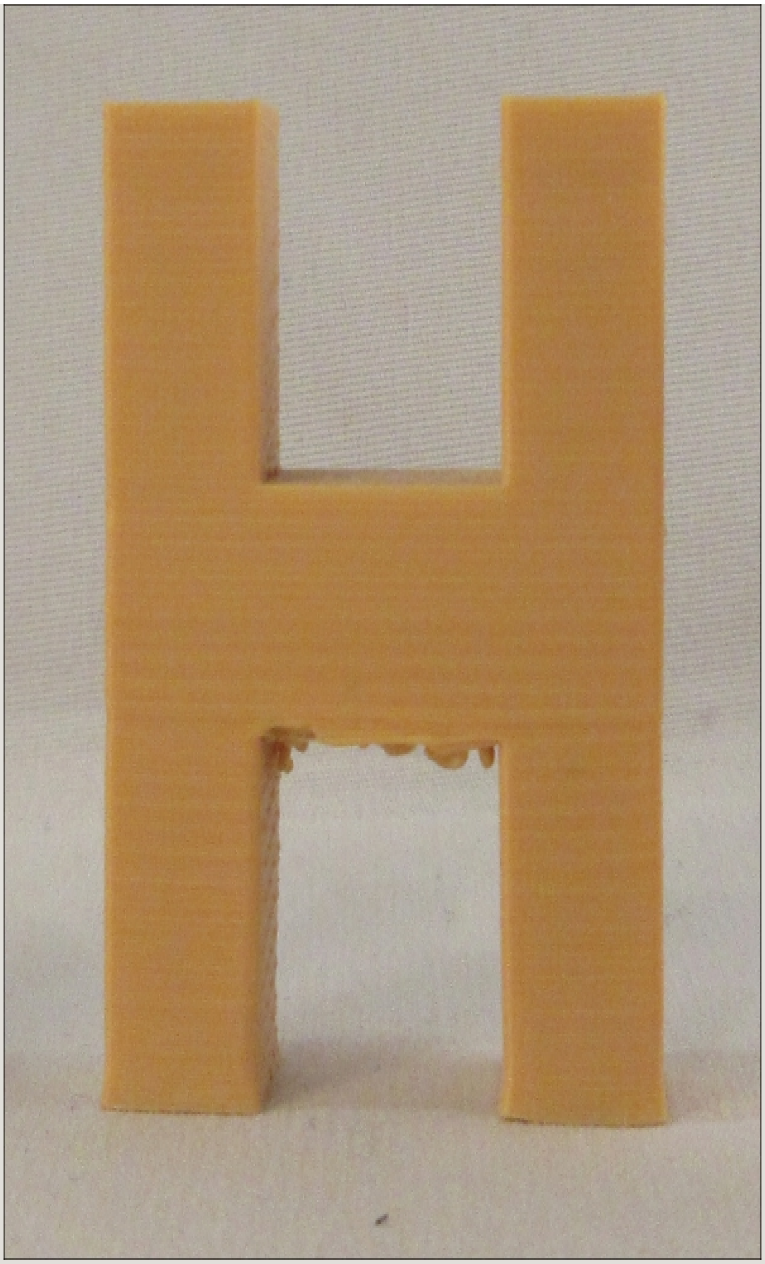
As the print gets to the part where the arms of the Y branch out, the change is gradual. It is possible to have the current layer slightly larger than the previous one, provided the overhang is gentle. Generally, a 45-degree overhang is safe. Hence, a shape like the letter Y will successfully print standing up.

However, if the overhang is too great or too abrupt, the new layer will droop, causing a print

to fail. Some 3D printer owners pride themselves in pushing their overhang and have seen success with angles as steep as 80 degrees, but to be safe, keep your angles no more than 45 degrees.

H – bridging

If a part of the print has nothing above it but has something supporting it on either side, like a capital letter H standing up, then it may be able to bridge the gap when printing:



Use caution when bridging. The printer makes no special effort when making bridges; they are drawn like any other layer: outline first, then infill. As long as the outline has something to attach to on both sides, it should be fine. But if that outline is too complex or contains parts that will print in midair, it may not succeed. Being aware of bridges in the design and keeping them simple is the key to successful bridging. Even

with a simple bridge, some 3D printers need a little bit more calibration to print it well.

Hence, a shape like the capital letter H will successfully print most of the time because of bridging.

T – orientation

If you were to try to print a capital letter T standing up on the build platform, you would surely run into problems:

The top arms have far too much overhang to print successfully. Of course, the solution to this is simple: when designing, flip the T over or lay it down. In fact, every letter of the alphabet will print successfully if laid on its back, but the letter T illustrates this best. Sometimes, when designing a part for 3D printing, it's good to turn it around and orient it so that it prints well. Not every print needs to be printed in the same way it's going to be used.

Wall thickness

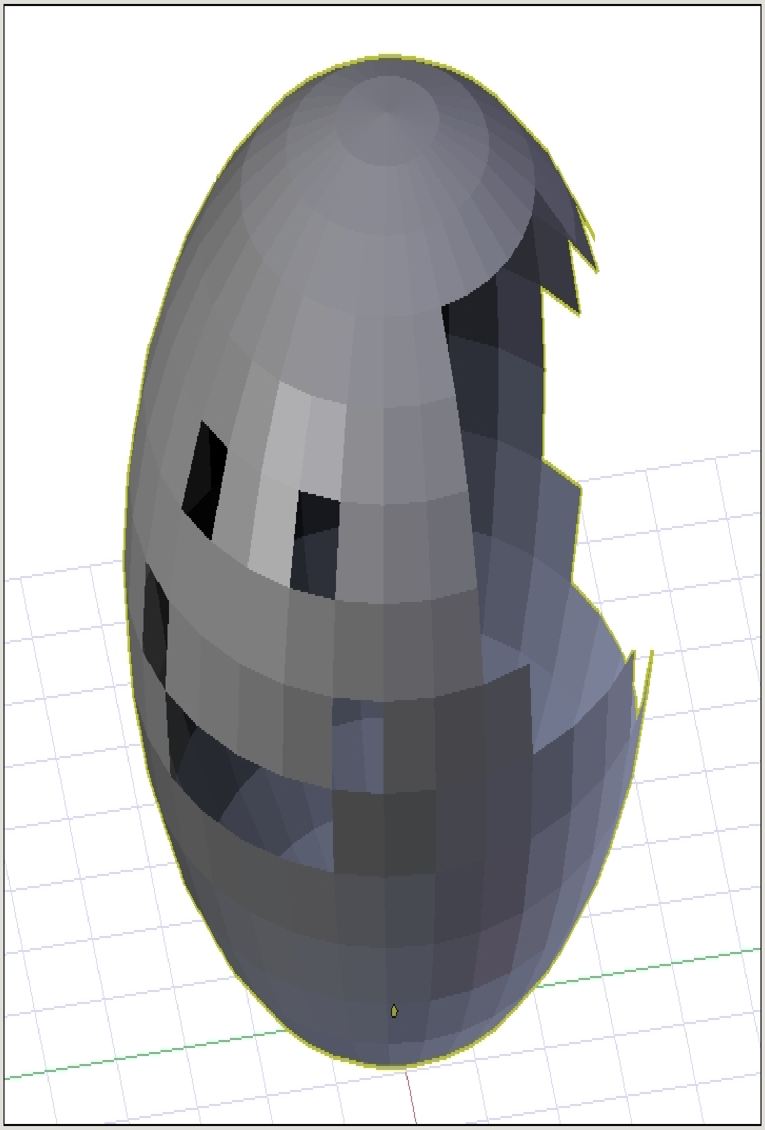
There is a minimum size of things that a 3D printer can print. This size is determined by the size of the hole in the nozzle, called the nozzle diameter. The most common nozzle diameter is 0.4 mm; however, most printers will not print a wall with a single extrusion thickness. They require that a wall be at least as thick as two nozzle widths, which in most cases means walls need to be at least 0.8 mm. However, because of the way slicers calculate outlines, 0.8 mm isn't just a minimum wall thickness—it's a target. For instance, if the wall is 1 mm thick, it won't be able to fill in the gap between the outlines, and there will be an air pocket. And while 0.4 mm is a very common nozzle diameter, it is not the only nozzle diameter, so a 0.8 mm wall may still be too thin for some 3D printers.

For thickness, it's best to err on the side of caution. A 2 mm wall is thick enough that slicers can use one or two outlines without conflict and still have room for a little infill, no matter the nozzle diameter. This will make solid prints that will succeed in almost all cases, and 2 mm is still fairly thin, allowing for considerable detail. Unless you are designing for a specific printer or planning to share your model with others, always make your walls a minimum of 2 mm thick to be safe.

Holes in models

Models for 3D printing must be closed, that is to say, they must have no holes in them. In a classic cartoon, there was a scene where bubbles were blown, but they were not bubble shaped. They were square, squiggly, and pink-elephant shaped. But no matter their shape, they were still

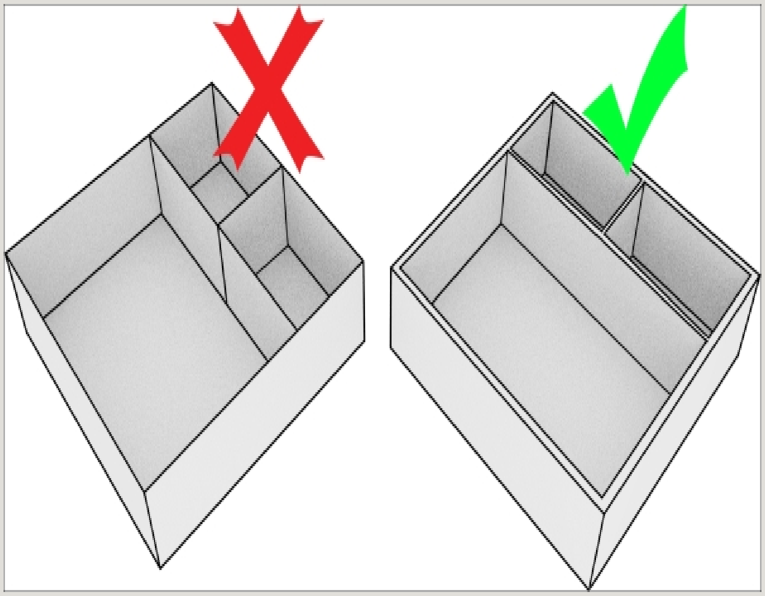
bubbles. If a hole developed in them, they popped. In the same way, models for 3D printing cannot have holes:



In mathematical terms, holes in models are included in a family of errors called non-manifold. Models for 3D printing must be manifold or else the slicer will have trouble telling what is supposed to be the inside and outside of the model.

In the same vein, a wall by itself, without an inside or outside, isn't printable because a 2D

wall has no thickness and doesn't describe a shape that can exist in real life. 3D prints must be part of a three-dimensional shape with a thickness, as described in the previous section.



Summary

3D printing is cool and allows the creation of fantastic and detailed objects without needing much interaction with people after the design is done. But designing for 3D printing is a lot like designing for any other type of manufacturing. It helps to know a bit about the process involved and design with that process in mind.

Fused filament fabrication 3D printing, or FFF for short, is one of the oldest, most mature, and cheapest forms of 3D printing, so this series will focus on designing for it. It involves melting a plastic filament and drawing the object layer by layer, with each layer sitting on top of the one below it.

Designing for the most effective FFF printing means thinking about overhangs and supports and about the parts of the prints that don't have anything underneath them when they print. To avoid needing supports when printing, it can help to remember the letters Y, H, and T when designing, in order to remember to consider gradual overhangs, bridging, and orientation. In addition, it's important to remember that details should be, generally, about 2 mm thick.

Now that the mechanics of 3D printing and how they affect design have been covered, the next chapter will deal with the specific software that will be used in this series.

Chapter 2. Beginning Blender

3D printers need 3D models to print. Those models don't just come out of nowhere. You will need imagination, a little time, and software to create 3D models for the things you want to print. Never have there been more software options for creating 3D models—professional and free options. In this series, the software of choice is called Blender.

This chapter will introduce Blender, how to set it up, and some basic and mid-level functionality. Knowing the content of this chapter will get you over Blender's infamous learning curve and provide the basic knowledge and reference material necessary to follow along with future projects. We'll cover these topics:

- Why Blender?
- Downloading and installing Blender
- The default view
- The best settings
- Object creation
- Navigating the view
- Transforming
- Controlling transformations
- Selecting
- The Edit mode
- Blender to real life
- Exporting an STL

Why Blender?

With so many options for 3D modeling software, why would Blender, a software designed to make 3D animations, be the most popular choice?

The price is right

First things first: you don't have to pay for Blender. It is offered free of charge. If it works for you, you always have the option to donate, but Blender doesn't do anything to force this point. It is free now and forever.

Blender is comprehensive

While it's true that Blender is designed for animation because it covers everything from a blank canvas to a finished animation, it contains the ability to model objects, and it's one of the most robust suites of modeling tools anywhere. Learning Blender means that you may never need to learn another 3D modelling software.

It's getting better all the time

Blender is in constant development. If it doesn't have a feature you need, chances are that it may one day. Blender's developers are constantly responding to their audience.

But Blender isn't perfect

Despite Blender's advantages, it has a well-earned reputation for having a difficult-to-overcome learning curve. It's had a long and organic development cycle, which left it with a default user interface that isn't intuitive compared to most other software.

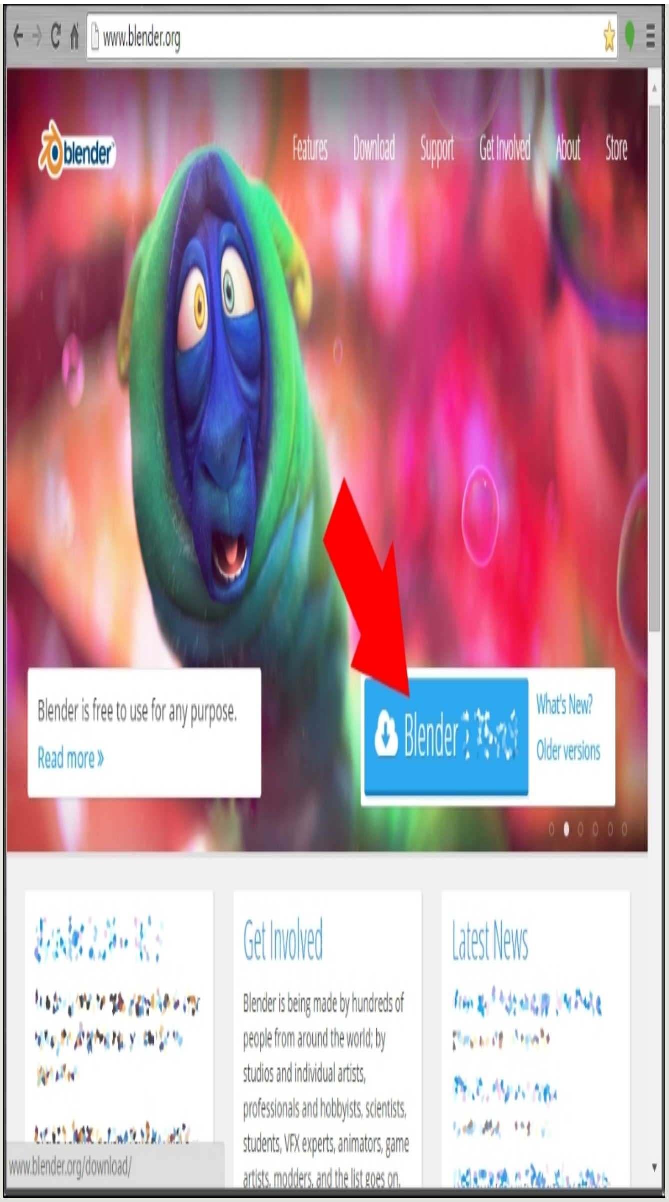
But Blender is also very configurable and, with a few simple settings, it can be made much easier

for the beginner to use. The rest of this chapter will help get you past Blender's short comings so that you can start developing awesome 3D models.

Downloading and installing Blender

The first thing that needs to be done is downloading and installing Bender. Follow these steps:

1. On your PC or Mac, open a your web browser and go to http://www.blender.org.
2. Locate the **Blender** download button on the main page for the latest version of Blende
3. r and click on it:

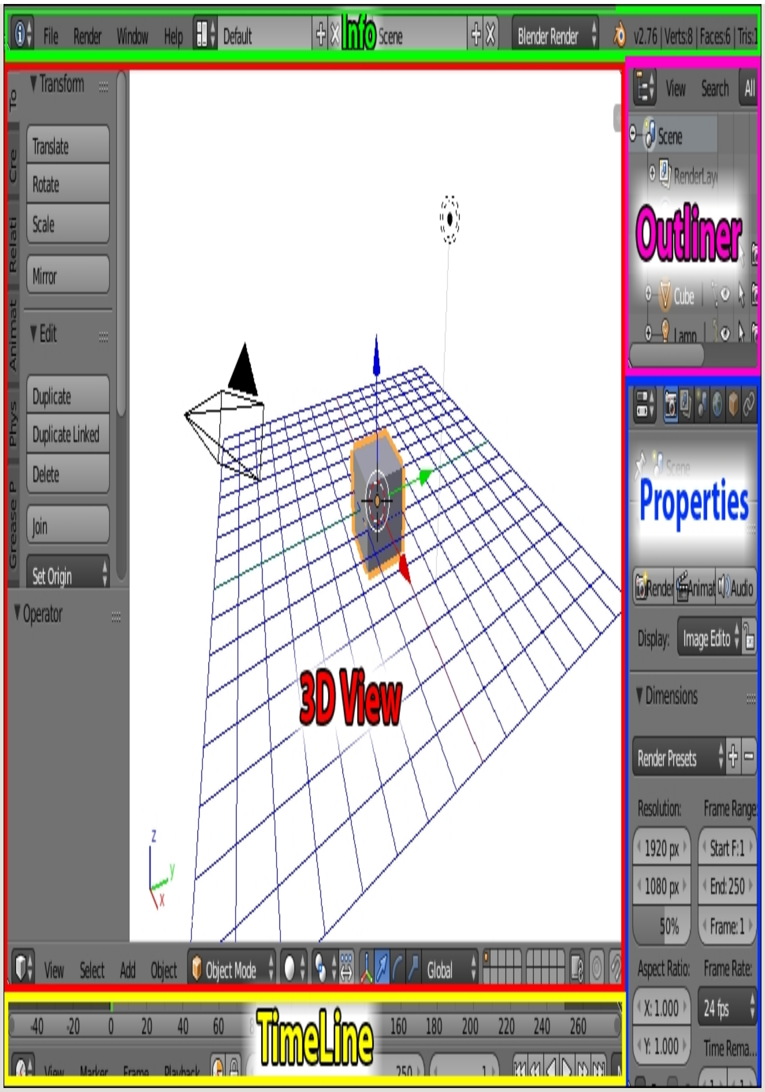


4. Scroll down and find **Download** under **Blender** and click on it. If you're on Windows and unsure, just choose the `MSI` package option.
5. When the installer finishes downloading, run it.
6. Follow the prompts to install **Blender**.
7. When the installer is finished, run **Blender**. Click anywhere to close the splash screen.

Blender is now installed and ready to use.

The default view

Blender's interface is made up of many smaller windows called **panels**. There are many different panels available in Blender. Like most things in Blender, the panels are completely configurable. Panels can be added or removed as needed, and panel layouts can be saved and switched among easily. For simplicity, the default view—the way Blender is presented the first time it loads up—will be used throughout this series. It provides most of the necessary functionality:



NOTE

For the most part, the screens shown in this book series will look similar to the default Blender screen. The major change will be to the background color of the 3D View, a choice made to make the illustrations more compatible with printing.

Here's an explanations of the different panels in the default view:

- **The Info panel**: Located across the top of the window, the **Info** panel has many of the menu options in most programs, such as **File**, **Window**, and **Help**. It also has **Layout** settings, **Scene** settings, and **Renderer** options, but we won't be using them much. Finally, there are specific details about the version of Blender, the current scene, and selected objects. If the **Info** panel is cut off,

bringing the mouse pointer over to the panel and using the scroll wheel will bring the information into view.

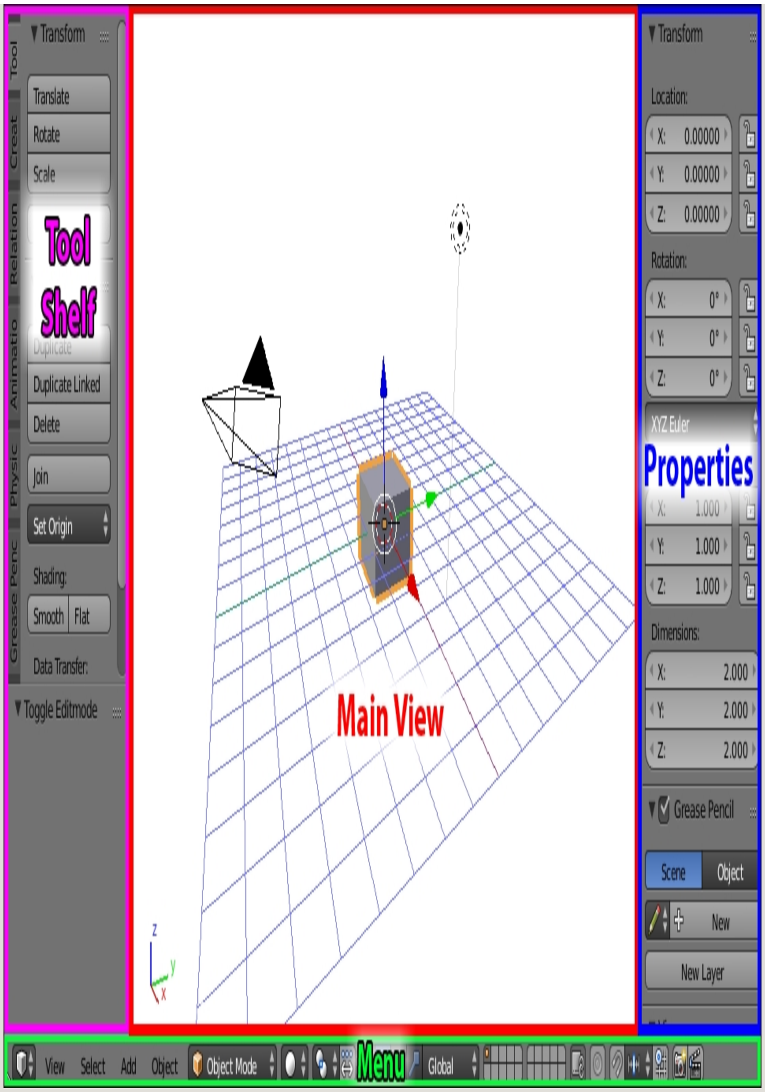
- **The Outliner**: Located in the upper-right section of the window, the **Outliner** contains the list of all objects.
- **The Properties panel**: Located in the lower-right section of the window, the **Properties** panel is broken into many tabs related to the currently selected object. Which tabs and properties are available will change depending on what is selected. If the tabs are cut off, bringing the mouse pointer over them and using the scroll wheel will bring them back into view.
- **The Timeline**: This panel is largely unnecessary for the purposes of this book series, so it can be removed or ignored.
- **The 3D View**: Taking up most of the screen, the **3D View** is where most things will be happening, and it consists of many smaller parts itself.

Each of the panels has its own keyboard shortcuts. In order to use them, the mouse pointer must be over the panel. If a keyboard shortcut isn't working, it's probably because the mouse pointer isn't in the right place.

The 3D View

Because it's so complex and important, the **3D View** will be given some special attention.

The **3D View** is where most of the action takes place, and it has the most visual feedback of the work. The **3D View** has its own menus and panels:



Here's what the panels do:

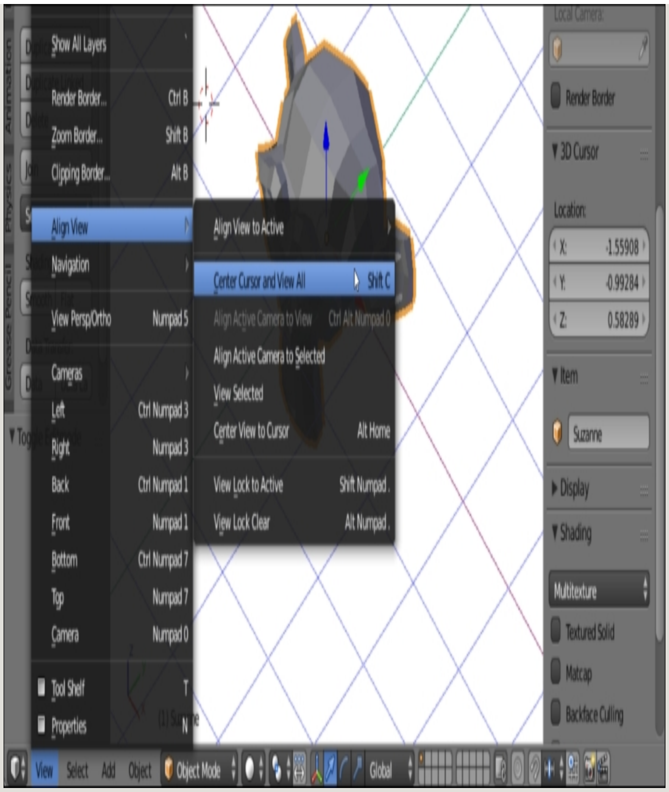
- **Menu**: At the bottom of the **3D View** is its **Menu**. It contains options and commands specific to the **3D View**. If the menu is cut off, bringing the mouse pointer over the panel and using the scroll wheel will bring the cut-off information into view.
- **Tool Shelf**: On the left-hand side is the **Tool Shelf**. It contains a tabulated set of buttons that can be used to do many useful things and change depending on the object selected. The bottom half of the **Tool Shelf** contains operator options. Any time an operation is performed, this area will be populated with options for that operation that can be edited until the next operation is performed. The **Tool Shelf** can be hidden or revealed by using the **3D View** menu and selecting **View** | **Tool Shelf** or by pressing *T* key on the keyboard.

- **Properties**: On the right-hand side is the **Properties** panel. Hidden by default, this panel contains information about the currently selected object as well as other options about the scene in general. The panel can be hidden or revealed by using the **3D View** menu and selecting **View** | **Properties** or by pressing *N* key on the keyboard.
- **Main View**: Of course, in the middle is the view of the current scene with all the visible objects.

The 3D cursor

In the **3D View**, there is a little red-and-white circle that starts in the middle of the 3D space. New objects will be created wherever the 3D cursor is located, and it's very easy to move the 3D cursor accidently, so it'd be good to know how to put the 3D cursor back in the middle and how to move it on purpose. The 3D cursor can be manipulated by:

1. Clicking the right mouse button in the **3D View** to move the 3D cursor wherever it is clicked in a plane relative to the view, which can be unpredictable.
2. Navigating to **View** | **Align View** | **Center Cursor and View All** or pressing *Shift* + *C* on the keyboard while in the **3D View** to put the 3D cursor back in the middle quickly. This is the easiest way to fix the 3D cursor if it gets moved unexpectedly.

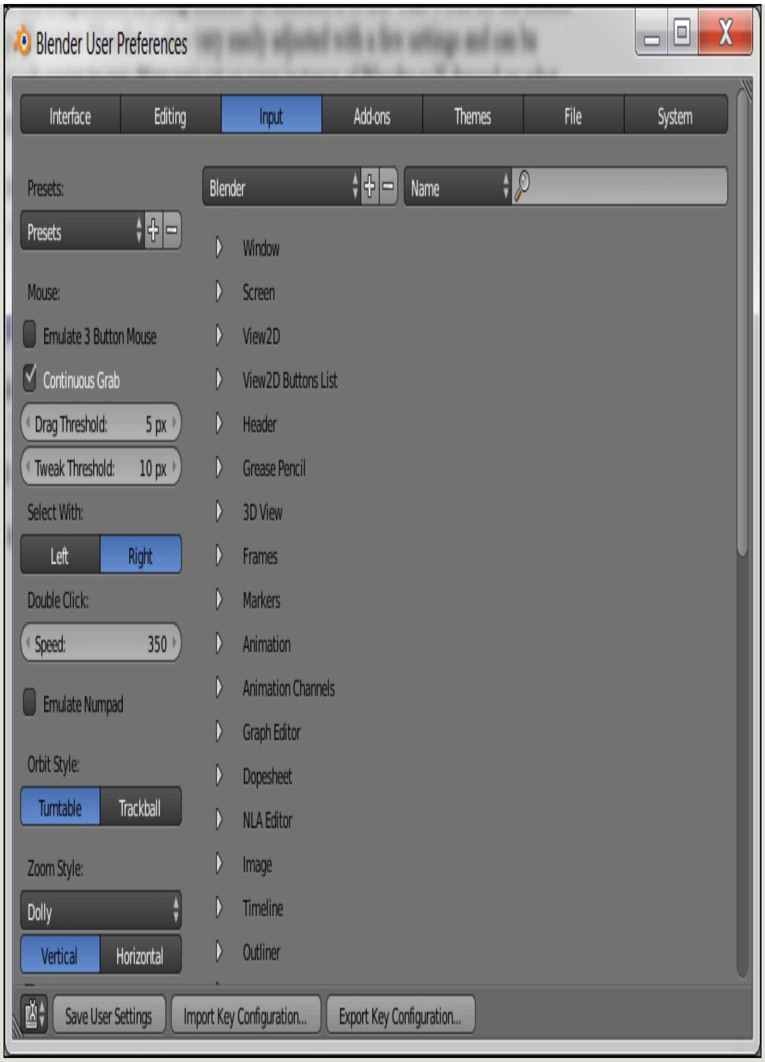


3. Typing the 3D cursor's X, Y, and Z coordinates in the **3D View** properties.

The best settings

Blender has a reputation for being difficult and unintuitive to use. This is true for the default settings. However, Blender is also very easily adjustable with a few settings and can be made much easier to use. How you set up your instance of Blender will depend on what your setup is like.

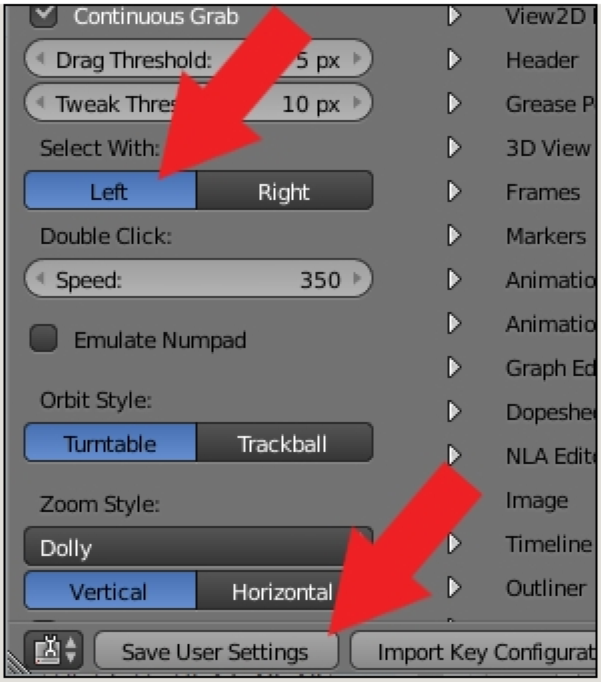
To access the settings, in the top menu select **File** | **User Preferences**. In the Blender **User Preferences** panel that comes up, select the **Input** tab button:



A scroll-wheel mouse and number pad

The recommended setup for Blender is to use your keyboard and a mouse with a scroll wheel. In this case, there is only one setting that is recommended to be changed from the default:

1. Click on the button that says **Left** under the words **Select With**.
2. Click on the button that says **Save User Settings**:



Changing this one setting will make Blender much more intuitive to use.

Because the default is for the right mouse button to be the select button, and some may prefer to keep this default, this book will refer to whatever option is chosen as the select mouse button, and the other button as the not-select mouse button. While still a bit confusing, it will help users who miss this section.

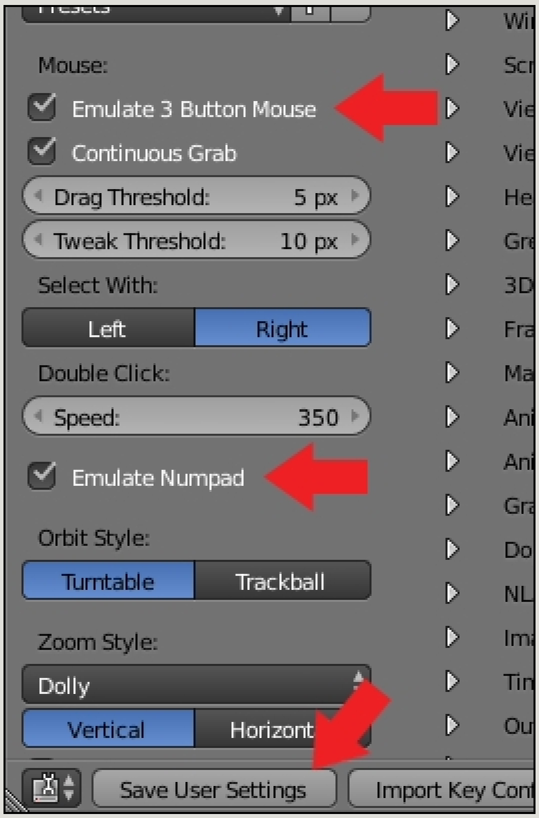
With a scroll-wheel mouse, the scroll wheel can be depressed for a middle mouse button click. Blender uses this middle mouse button to manipulate the view.

If you have a Numpad but no scroll-wheel mouse, it is still recommended to use these settings because the interface will be much more intuitive, although you will be sacrificing some functionality that the middle mouse click offers. The choice is yours whether it's worth having a more intuitive interface.

A laptop with a touch pad and no number pad

On a laptop with a touchpad (with no middle click) and no number pad, both very important for navigating the view, select the following settings:

- **Emulate 3 Button Mouse**
- **Emulate Numpad**
- Click on the button that says **Save User Settings**



With this setup, most of Blender's functionality is available to laptop users, although these settings are less intuitive. With these settings, you will need to use the right mouse button to select objects, *Ctrl* + right mouse button as a middle mouse button to change the view, and the number keys across the top will perform the function of the number pad on a regular keyboard.

With Blender set up, it's time to start learning to use it. Close the **User Preferences** window.

> **NOTE**
>
> Blender users tend to use keyboard shortcuts for almost everything. It's recommended that you use Blender with one hand on the mouse and the other on the keyboard. Learning and using keyboard shortcuts will speed up your process and, with just a little practice, will become second nature. Going forward, all methods of accessing commands will be taught. Try to practice using the keyboard shortcuts.

Object creation

Most tutorials for Blender start with navigating the **3D View**. But in Blender, the default scene is kind of boring for this, just a cube that looks the same from every angle. Instead, let's make something more interesting to look at.

First, let's look at how to clear the default scene:

1. Select all the objects in the scene by going to the **3D View** menu and choosing **Select** | **(De)select** all twice or pressing the *A* key twice. Everything in the scene should have an orange line around it. If not, do it again.
2. Delete everything by going to the **3D View** menu and choosing **Object** | **Delete** or pressing the *X* key.

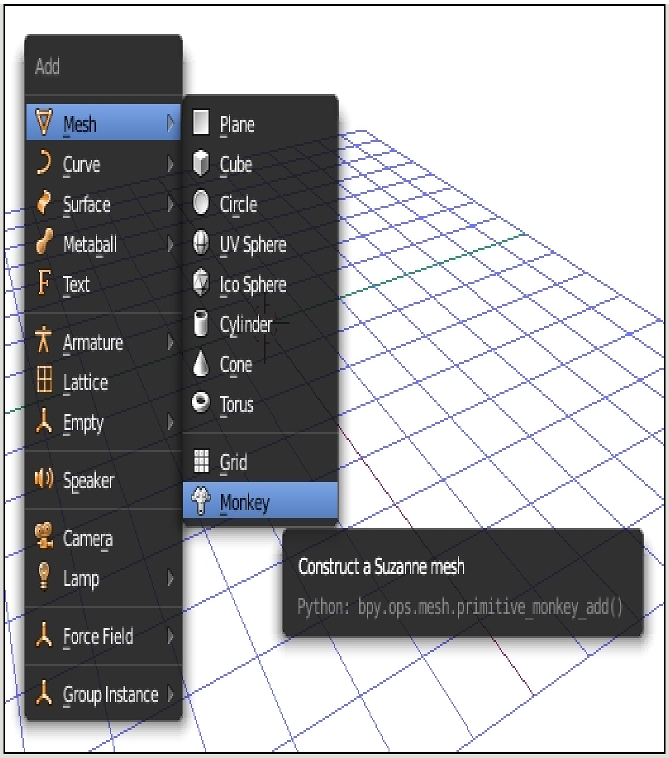
Your **3D View** should now have nothing but the grid, which can be thought of as the floor of the scene.

To create an object, go to **3D View** menu, choose **Add** or press *Shift* + *A*.

In the menu that pops up are all the basic objects that can be inserted into the scene. Blender offers many basic shapes that can serve as a starting point for anything you want to create. Sometimes, a basic object is all you need, and sometimes, the basic object needs to be modified.

In this case, add a **Monkey** object to the scene:

1. In the **3D View** menu, choose **Add** or press *Shift* + *A*.
2. Select **Mesh** | **Monkey** in the menu that appears.



The **Monkey** object is good when learning to navigate the view because it's easy to see which way it's facing, no matter how the view changes. So, with a monkey object in the scene, it's time to learn how to change your perspective.

Navigating the view

Because Blender is all about working in 3D, but computer screens are flat, it is important to know how to change how you're looking at something in Blender.

NOTE

All of the following commands can also be found in the **3D View** menu under **View** | **View Navigation**; however, since adjusting the view happens so frequently, it is recommended to learn the mouse and keyboard shortcuts instead of navigating menus to do this.

Rotating the view

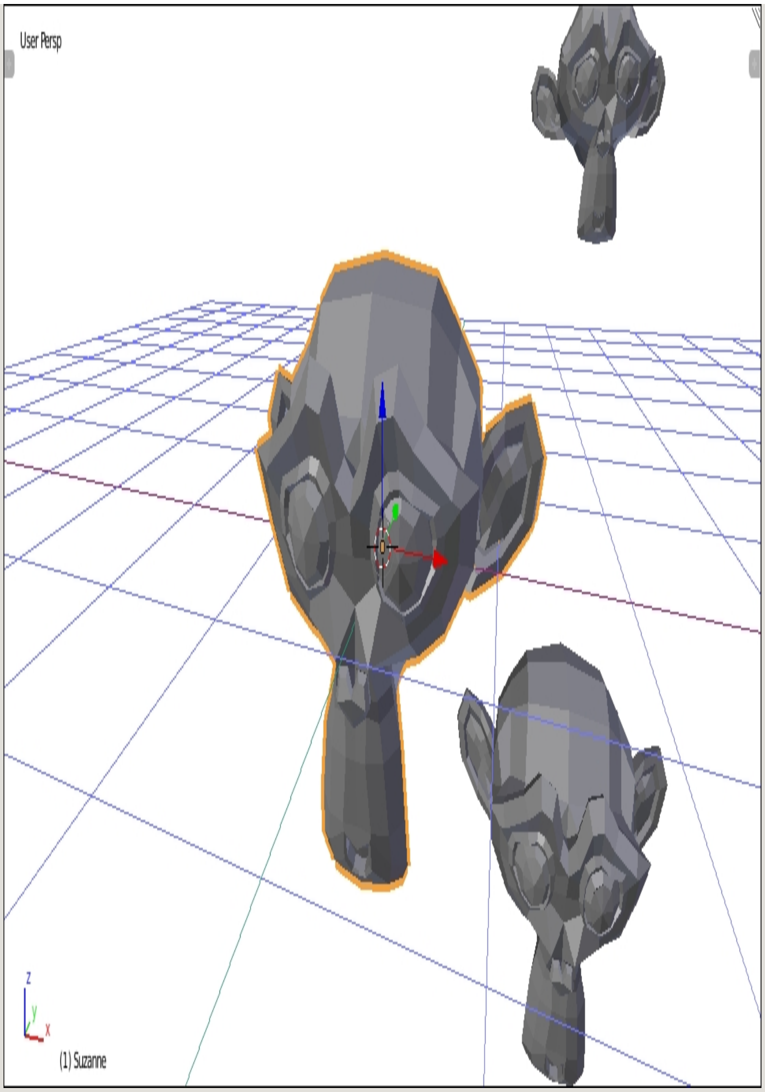
In Blender, you can change the angle of the view by:

- Clicking and holding the middle mouse button and moving the mouse pointer.
- Pressing *2* or *8* on the number pad to rotate the view up and down.
- Pressing *4* or *6* on the number pad to rotate the view left and right.

The point at which the view is rotating can change. Follow these steps to center the view rotation on a specific object:

1. Select the object.
2. Press the . (period) key on the number pad.

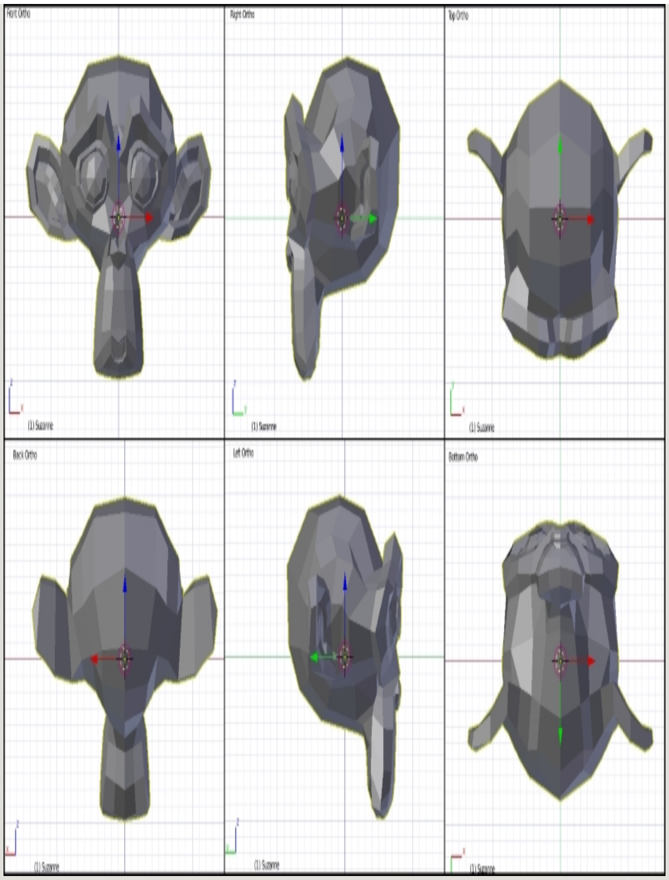
The selected object will fill the view and all view rotations will now center on that object:



Jumping to rotation

Blender has six set views that can be quickly jumped to at any time:

- Press *1* on the number pad to jump to the front view
- Press *3* on the number pad to jump to the right view
- Press *7* on the number pad to jump to the top view
- Press *Ctrl* + *7* or *9* on the number pad to jump to the bottom view
- Press *Ctrl* + *1* on the number pad to jump to the back view
- Press *Ctrl* + *3* to jump to the left view



Panning the view

Panning the view means moving without changing the rotation, like moving your head while it's still pointed in the same direction. To pan the view, follow these steps:

- Press *Ctrl* + *4* on the number pad to pan left
- Press *Ctrl* + *6* on the number pad to pan right
- Press *Ctrl* + *8* on the number pad to pan up
- Press *Ctrl* + *2* on the number pad to pan down
- Hold *Shift* while clicking and holding the middle mouse button and move the mouse

> **NOTE**
>
> Panning the view is one of the functions that changes the center of rotation. Remember: you can reset the center of rotation by selecting an object and pressing . (period) on the number pad.

Zooming the view

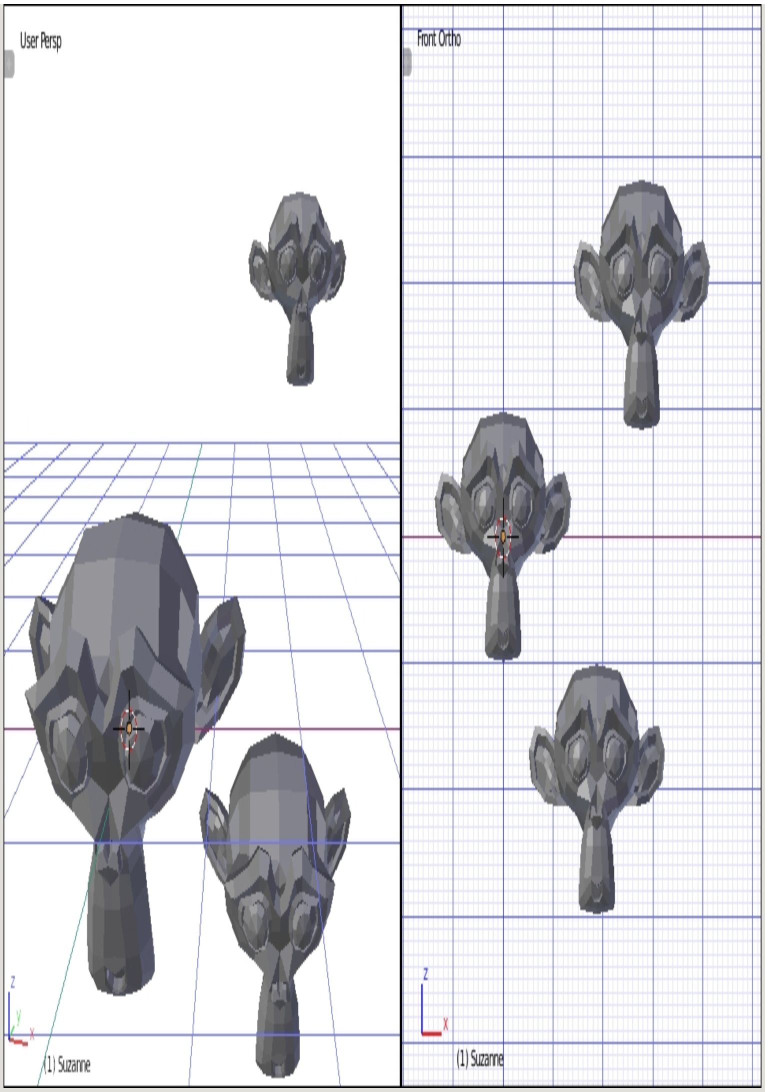
Zooming the view is moving closer to or farther from the object. To zoom the view, do one of the following:

1. Turn the scroll wheel.
2. Press Ctrl + middle mouse button and move the mouse.
3. Press *+* or – on the number pad.

Orthographic versus perspective view

To toggle between orthographic and perspective views, press *5* on the number pad.

The words **Persp** or **Ortho** can be seen in the upper right-hand corner of the **3D View**, indicating which view is being used:



There are two ways of looking at the **3D View** on a computer. Perspective is the default, and more closely resembles how things will look when viewed through a camera or our eyes: closer objects look bigger; farther objects look smaller. Orthographic makes everything the same size, no matter how close or far it is. This makes it easier to compare objects and determine their location relative to one another.

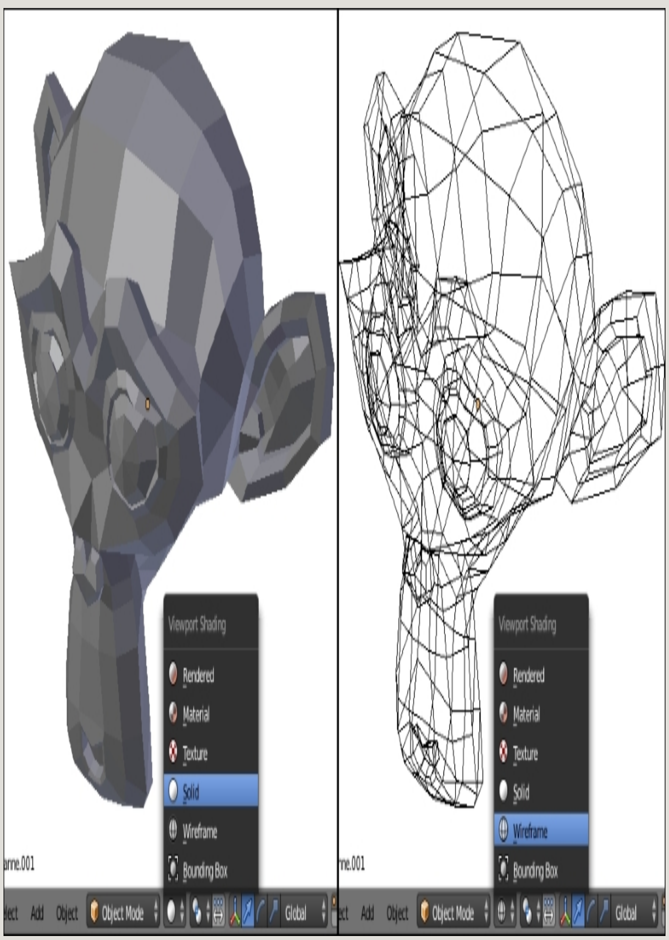
Because a lot of 3D modelling requires precision, orthographic is generally preferred; however, it can be confusing when two objects overlap exactly. For this reason, it's good to adjust the

view frequently and be sure that the action you're preforming is the action you think you're performing.

Wireframe and solid view

With 3D modelling tools, it helps to be Superman. Being able to see through objects can help when selecting and modelling things. You can switch between the **Wireframe** and **Solid** views using one of these methods:

1. Press *Z* to toggle between **Solid** and **Wireframe** view.
2. On the **3D View** menu, click on the **Method to Display** popup and choose **Solid** or **Wireframe**.



Wireframe mode is very powerful, especially in **Edit** mode, but it can be very confusing, especially as objects get more complex.

Wireframe mode allows objects behind the objects in the foreground to be selected without adjusting the view. Get used to switching between **Solid** and **Wireframe** mode frequently.

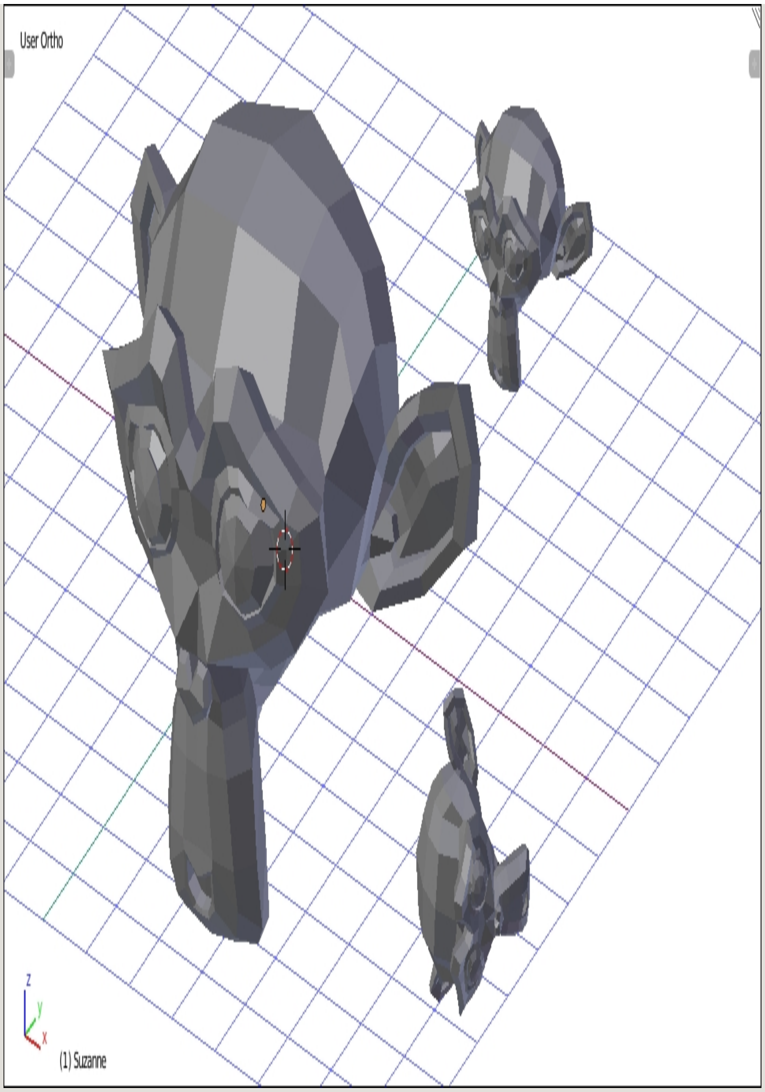
Transforming the object

In Blender, transforming an object changes the size, direction, or location of a thing without changing its shape. There are three basic transformation commands that are used frequently: `Grab` and `Move`, `Rotate`, and `Scale`.

To transform an object, be sure the object is selected, and then, from the **3D View** menu:

- Select **Object** | **Transform** | **Grab/Move** or press *G* on the keyboard to move the object
- Select **Object** | **Transform** | **Scale** or press *S* on the keyboard to scale the object
- Select **Object** | **Transform** | **Rotate** or press *R* on the keyboard to rotate the object

Then, move the mouse or use the arrow keys to transform the object. When the transformation has been accomplished, press *Enter* or the select mouse button to end the operation, or the transformation can be cancelled by pressing the not-select mouse button or the *Esc* key. The following diagram shows these object transformations:



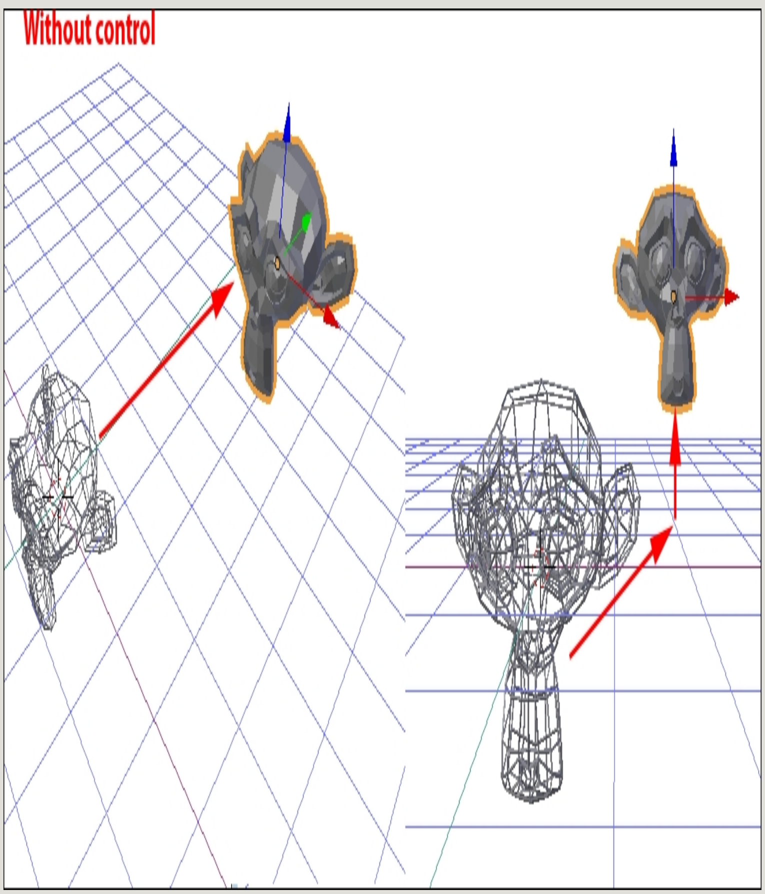
Transformations can also be undone after they're completed by pressing *Ctrl* + *Z*.

NOTE

Transform the **Monkey** added to the scene by moving, rotating, and scaling it. Get a feel for how these commands work. Then, try to move the **Monkey** to a specific place in the scene. When you think you have it, change your view to see whether it's really where you think it is.

Controlling transformations

By default, operators in Blender operate on a 2D plane tangential to the view. This is a fancy way of saying that without any additional controls, it can be hard to predict how a movement or rotation operation will work. For instance, moving something in a random view can include moving up and down more than expected. This effect won't be clear until the view is changed and the transformation is inspected from a different angle, as shown here:



If you can't predict how operations will work, it can be hard to make the things you want. So it is

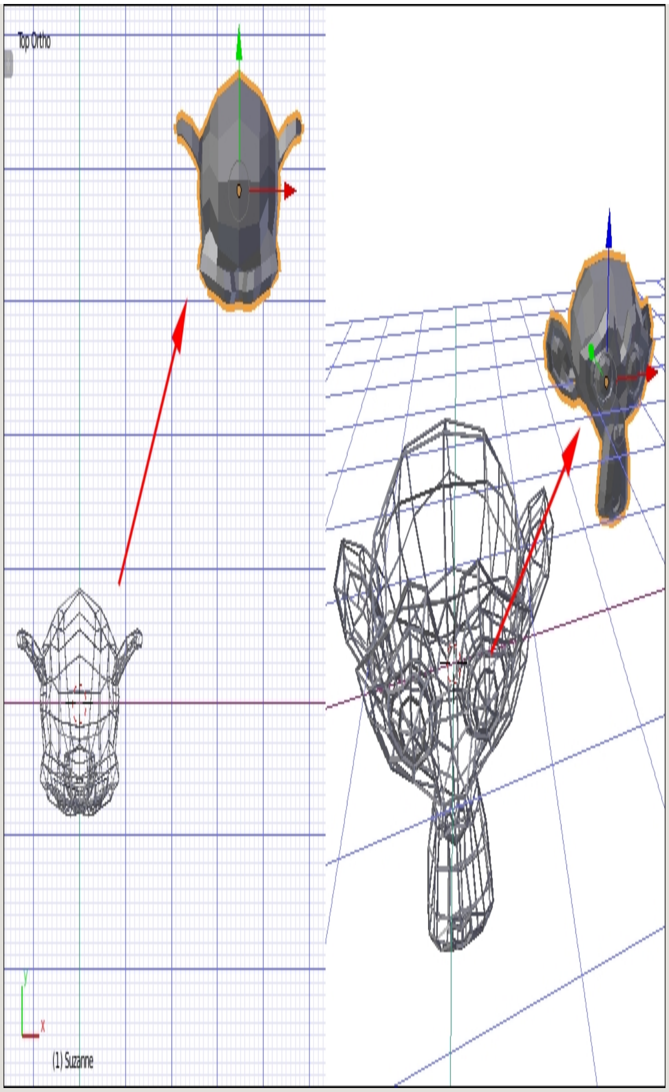
very important to be able to control transformations.

There are two main ways of controlling operators: **controlling the view** and **axis locking**.

Controlling the view

The first way to control the operation is by controlling the view. By default, operators depend on the view, so by controlling the view, you can control the action. For instance, an object, when added to a scene, is exactly halfway through the grid plane of the world by default. If you want to make sure the object stays halfway through the grid plane when moving, follow these steps:

1. Select a newly added object.
2. Switch to **Top Ortho** view by pressing *7* on the number pad.
3. Grab and move (*G*) the object around.



Adjust the view to see how exactly how the object moved. When moved while in the top view, the object stays on the grid plane. This is because from the top view, only forward, backward, and side-to-side motion is possible.

In the same way, moving objects while in the side views will only move them forward and back and up and down, and while in the front or back view, they will only move side to side and up and down.

Likewise, rotating depends on the view. When viewing from the top, rotation will be like, a top

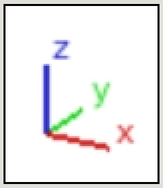
spinning around its middle. If viewed from the side when doing rotation, it will flip over its middle, and if viewed from the front, it will roll side to side around its middle.

Controlling the view is a quick and easy way of controlling movement and rotation, but always be sure to adjust the view to make sure things are happening the way you expect. However, to control scaling, as well as to control movement and rotation independent of the view, there is another way.

Axis locking

Blender has special commands for changing the behavior of operations. These commands are available while performing transformations.

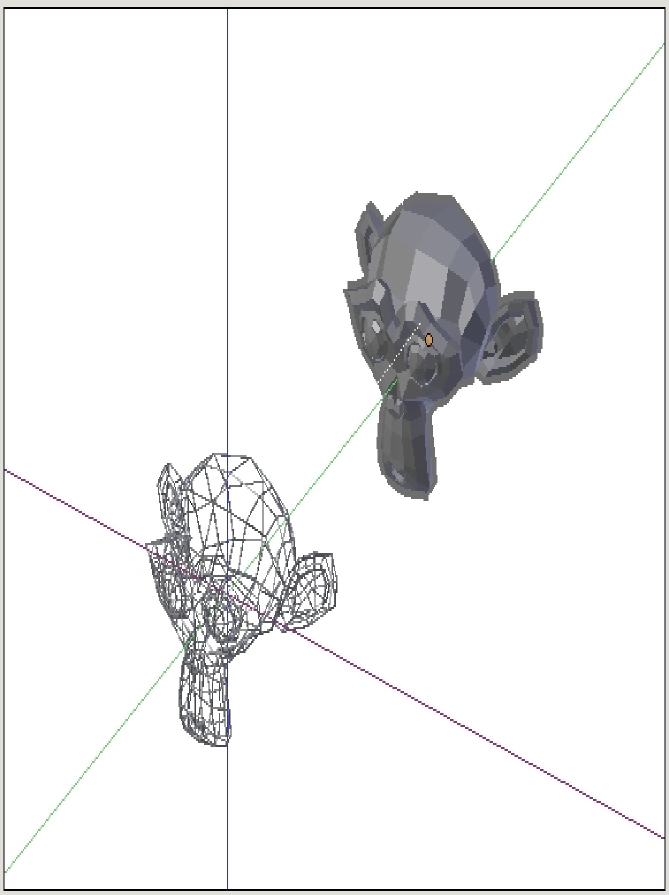
The first thing to understand is about the three axes that we'll be locking along. 3D refers to the three dimensions, or unique directions, that things happen in. The dimensions are called by the letters X, Y, and Z. There is an illustration in the corner of the 3D View that shows which way is which, and is always pointing in the right direction:



In Blender, and generally in 3D printing, X is side-to-side motion, Y is back-and-forth motion, and Z is up and down. In the previous screenshot, the X axis is red, Y is green, and Z is blue.

When performing a transformation, you can lock the transformation to the axis you want by starting the transformation as mentioned previously, and while moving the mouse or using the keyboard, doing the following:

- Pressing *X*, *Y*, or *Z* on your keyboard to lock the transformation to the desired axis
- Pressing *Ctrl* + *X*, *Ctrl* + *Y*, or *Ctrl* + *Z* to lock the transformation to all but the chosen axis
- Holding the middle mouse button and moving the mouse to choose an axis to lock the transformation to



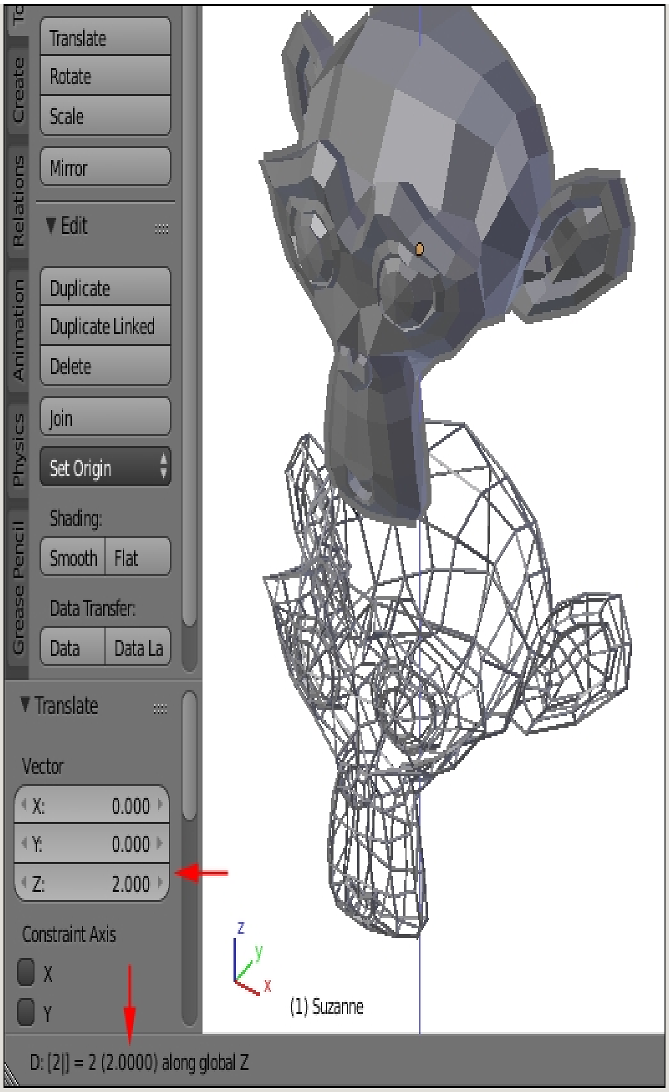
NOTE

Try out axis locking with movement, rotation, and scaling, and adjust the view to see what effect it has. Notice that with axis locking, the scale function has some additional abilities that aren't possible just by controlling the view. It actually only scales in the chosen axis. This can be very powerful while adjusting the shape of objects.

Precise transformation

Finally, under the category of controlling transformations, during transformation operations, the transformation can be precisely controlled by typing in a number that will relate to the operation being performed or edited afterwards in the operation properties in the tool box, on the left side of the **3D View**. In other words, to move something exactly two units up, for example, follow these steps:

1. Select the object.
2. Start the movement transformation (*G*).
3. Type *Z* to lock the movement to the Z axis.
4. Type *2* on the keyboard.
5. Press *Enter* or click the select mouse button to finish the operation.



Typing commands has different effects for different commands, as follows:

- When moving, typed commands are used to state the number of units an object will move along the selected axis. For example, *2* along the Z moves it two units up, and -2 along the Z moves it two units down.
- When scaling, typed commands specify the scale factor. 1 means no change, 2 means twice as big, and 0.5 means half size.
- When rotating, typed commands specify clockwise degrees. For instance, 180 turns it around backwards.

Typed commands can include negative numbers and decimals and can be edited with the *Backspace* key. While performing a

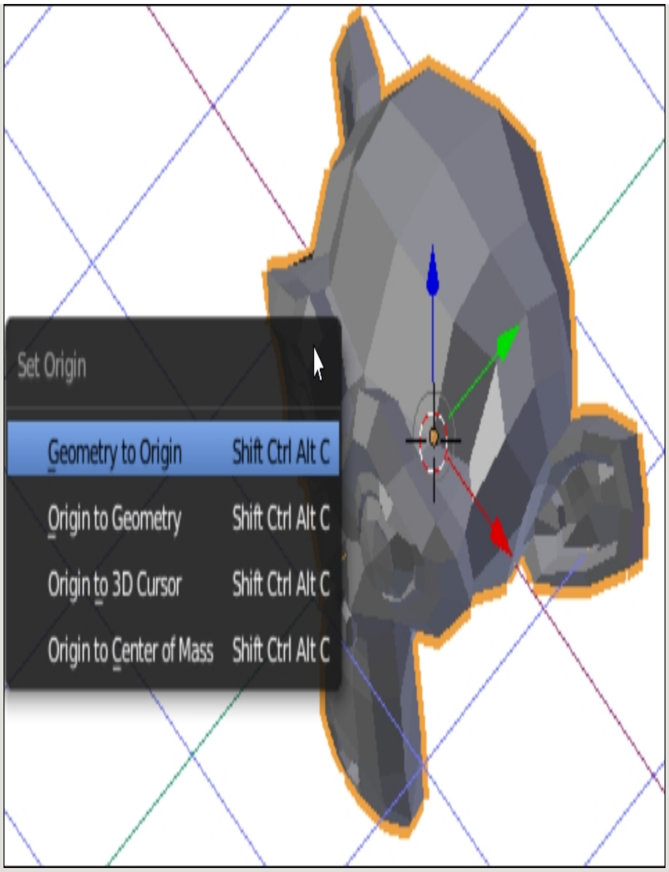
transformation, the menu for the **3D View** changes to a description of the transformation, including the typed units, which can be useful when editing your typed commands.

Origin manipulation

Objects in Blender have an origin. Origins are depicted as a dot and start out in the middle of the object. Individual object transformation commands take place relative to the object's origin. It's possible that the origin can be moved as a result of editing, which can cause unexpected results when rotating or scaling the object. The origin can also be moved on purpose to control the effect of modifiers.

The origin can be reset using the origin controls. The origin controls can be found by one of the following methods:

1. In the **3D View** menu, choose **Object** | **Transform** and find the origin controls.
2. In the **Tool Shelf**, find the **Set Origin** dropdown and choose the desired option.
3. On the keyboard, press *Ctrl* + *Shift* + *Alt* + *C*.



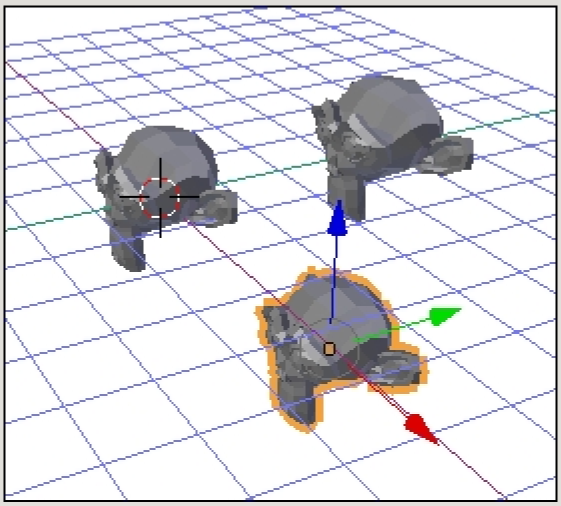
These are the commands to manipulate the origin:

- **Geometry to Origin**: Move the object so that its middle is now wherever the origin was located.
- **Origin to geometry**: Move the origin to the middle of the object. This is the most common option chosen.
- **Origin to 3D Cursor**: Relocate the origin to where the 3D cursor is.
- **Origin to Center of Mass**: Calculate the center of mass of the object and move the origin there.

Duplicating objects

Blender can duplicate existing objects. This is very useful and can speed up making things. To duplicate an object, follow these steps:

1. Select the object to be duplicated.
2. In the **3D** menu, choose **Object** | **Duplicate** or press *Shift* + *D*.
3. Move the mouse or use the arrow keys to place the duplicate (axis-constraining commands also work at this point).
4. Press *Enter* or the select mouse button.



Object selection

Naturally, Blender is capable of selecting more than one object at a time. Blender has many tools to help when selecting objects. With multiple objects, the transformation commands work on all the selected objects at once. This is a powerful way of controlling transformations while keeping objects in relative positions to each other.

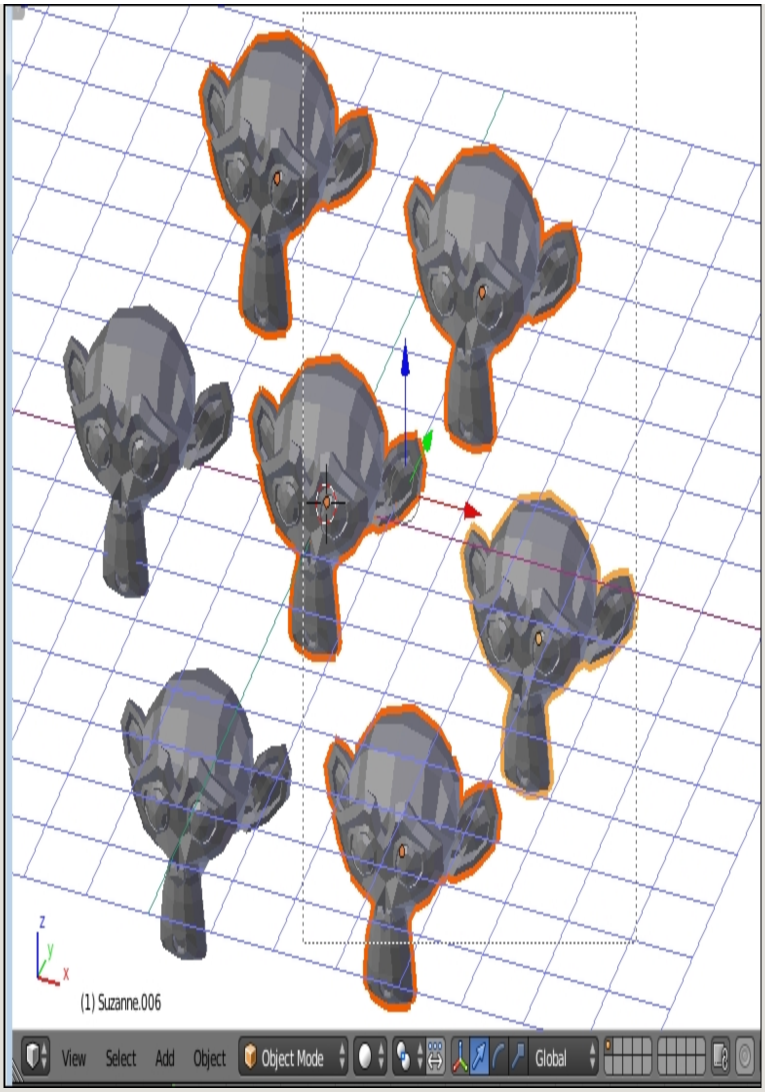
> **NOTE**
>
> To practice selection, there need to be multiple objects to select. Create a scene, add an object, and then use the duplicate (*Shift* + *D*) command to create many objects from the one. Spread them around the scene so that they can be selected with the following tools.

Shift select

One way to select multiple objects at once is to hold down the *Shift* key while clicking on the desired objects one at a time. To deselect an object, it must first be made active. The active object is highlighted in a different color. Then it can be clicked again while holding the *Shift* key, and it will be removed from the selection.

Border select

Another way to easily select multiple objects in a scene is to use the border select command and draw a box around the objects you want to select. To border select, choose **Select** | **Border Select from the 3D View** menu or press *B* on your keyboard. Then, click and hold down the select mouse button, move the mouse pointer, and release the mouse button:



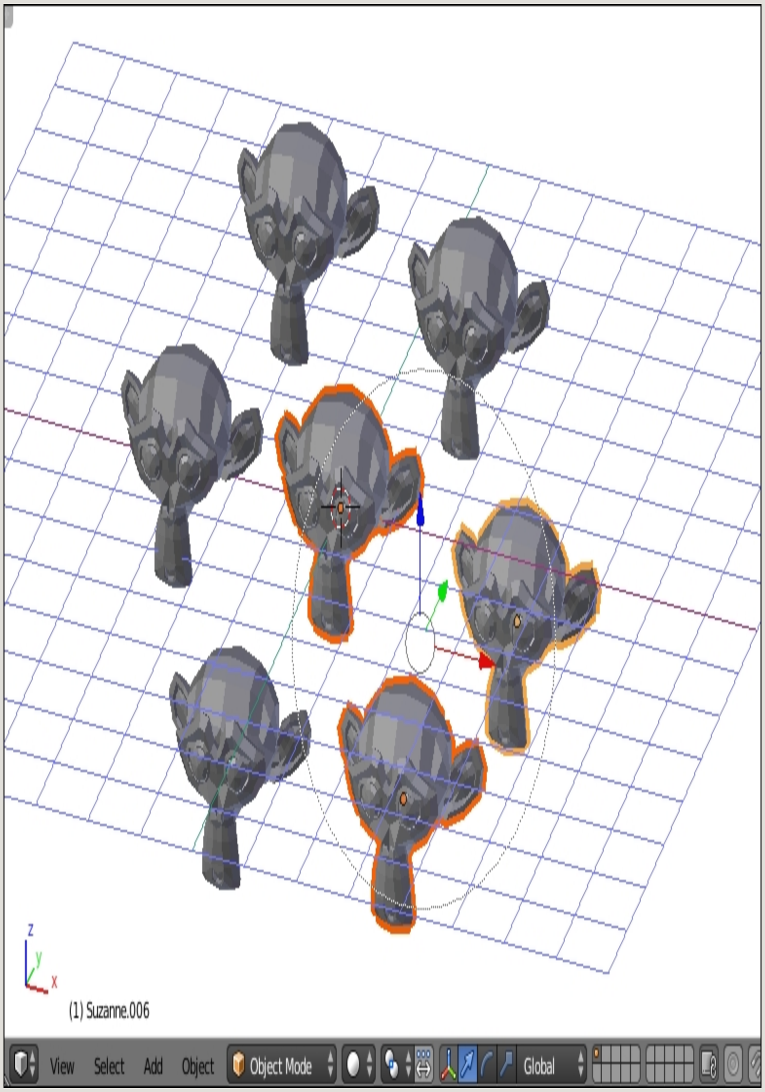
If even a small part of an object is included in the box, it will be added to the selection.

To deselect objects with border select, use the middle mouse button when drawing the box.

Circle select

Circle select is another powerful tool for selecting multiple objects. To circle select, choose **Select | Circle Select from the 3D View** menu or press *C* on the keyboard. A circle will appear around the mouse pointer. Use the scroll wheel or *+* and – on the Numpad to increase or decrease the size

of the circle. Then, click or click and hold the select mouse button, and everything inside the circle will be selected. Use the middle mouse button to deselect:



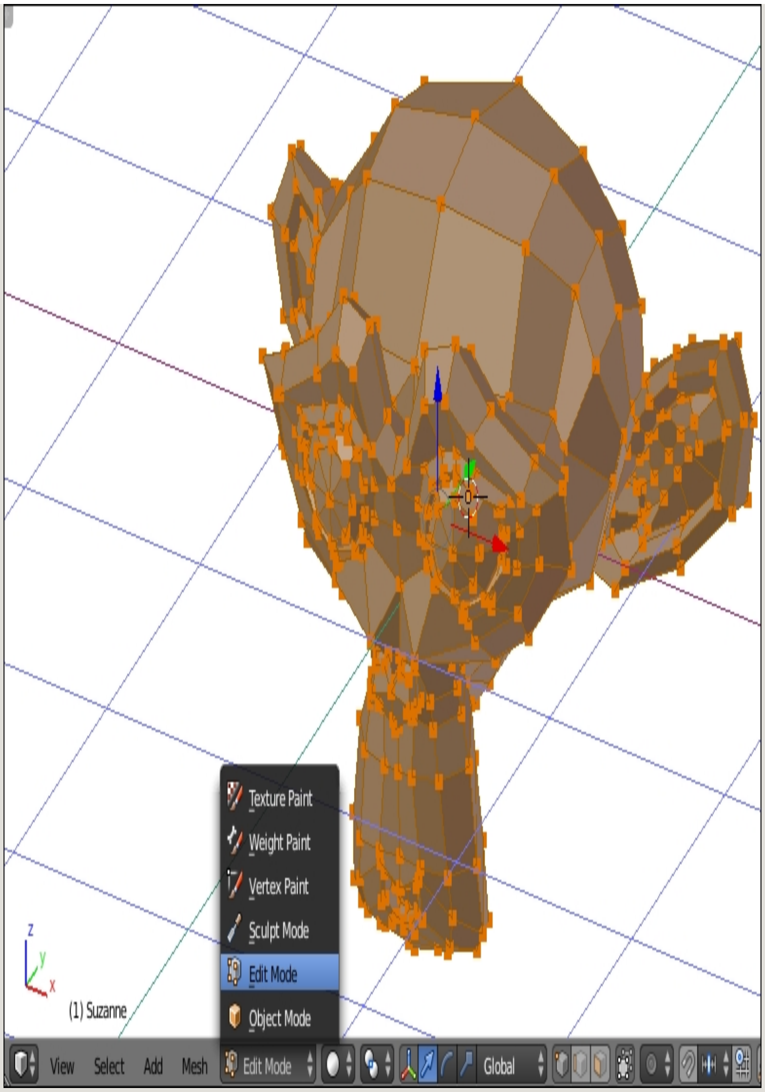
Circle select only adds or removes an object from the selection if its center or origin is inside the circle when selecting.

The Edit mode

In Blender, the **Edit** mode allows more access to the shape of a single object so that it can be manipulated in order to change its shape. To enter **Edit** mode, use this method:

1. Select an object.
2. In the **3D View** menu, locate the mode pop-up menu and select **Edit Mode** or press *Tab* on your keyboard.

In **Edit** mode, the **3D View** menu, **Tool Shelf**, and **Properties** all change, adding new functionality only available in **Edit** mode:



Parts of objects

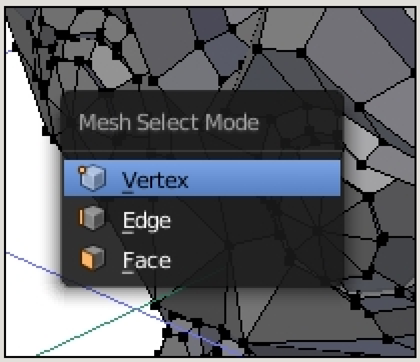
In **Edit** mode, objects are broken down into three parts:

- **Vertices**: Points in three-dimensional space. Vertices don't have any shape by themselves
- **Lines**: Two points are connected with a straight line between them
- **Faces**: Three or more lines can be connected to make a face

There are many ways to think about vertices, lines, and faces. For instance, if making a kite, the vertices are the joints, the lines are the

sticks, and the bits of paper are the faces. If the location of the vertices is moved, the shape of the kite will change. It's the same with a 3D object. Editing by vertices, lines, or faces will affect the rest.

The **Edit** mode is by default in vertex select mode, meaning any selection or transformation is applied to vertices, but it's easy to switch by locating the vertex, line, or face select buttons in the **3D View** menu or by pressing *Ctrl* + *Tab* and selecting the desired option from the menu that pops up:



Try out **Select** mode. Select groups of vertices, edges, and faces, using circle and border select. Try moving, rotating, and scaling them and see how it effects the shape of the object. Try simple basic shapes such as cubes and circles. See how **Wireframe** mode affects selection. Notice how the other objects in the scene can't be accessed.

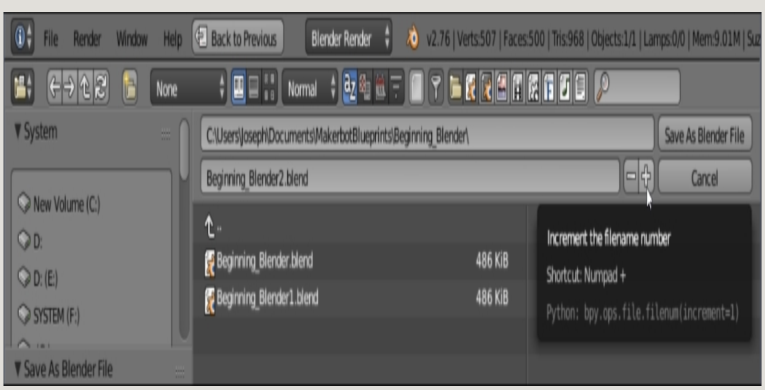
A lot of what happens in the projects will happen in **Edit** mode. But the individual projects will better teach you how to use it.

Incremental saving

It is always a good idea to save your work frequently. To save your work in Blender, choose **File** | **Save** from the info panel or press *Ctrl* + *S*. Then, navigate to a chosen folder or directory, give the file a name, and click on the **Save As Blender File** button or press the *Enter* key.

It is recommended that each Blender project gets its own folder and that all projects be saved in a location that will be easy to find later.

It is also a good idea, while learning especially, to give each version of the file you save a slightly different name. This way, there is a history of the work done and it's easy to go back in the case of a mistake that isn't discovered immediately. This is called incremental saving and is simple to do in Blender. Simply choose **File** | **Save As** or press *Ctrl* + *Shift* + *S* to get to the save menu. If the project has been saved previously, it should already have a name. Next to the filename, there are plus and minus buttons:



By clicking on those buttons, the filename will have a number attached to it, and that number will be increased every time the plus button is

clicked. Then, click on **Save as Blender File** or press the *Enter* key to save the file with a new filename.

Blender to real life

By default, Blender units don't make any attempt to relate to real-life measurements. However, after exporting a mesh, the slicing software will interpret the Blender units as millimeters, generally. So, it is good to think of Blender units as millimeters. This means that default objects in Blender are 2 mm across when they're added, which is fairly small.

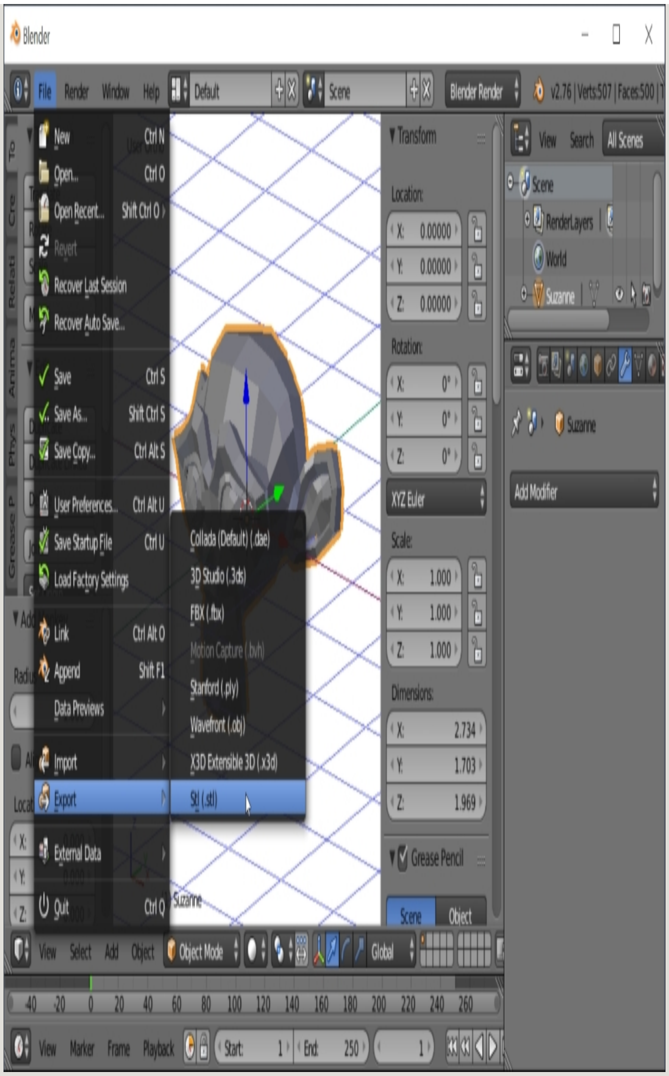
Remember that one blender unit will be 1 mm when the file is used to 3D print an object.

Exporting an STL

Before they can be used in a 3D printer, 3D models created in Blender have to be changed to a file that the 3D printer can use. Blender's default file format isn't readable by 3D printers and can sometimes contain additional information that the 3D printers don't need.

Most 3D printers use a file format called **STereoLithography** (**STL**), which contains just the final shape of the object. To export a finished model to an STL for 3D printing, follow these steps:

1. Select the model or models to be exported.
2. From the **Info** panel, choose **File** | **Export** | **Stl (.stl)**.
3. Navigate to a chosen folder or directory.
4. Give the file a name.
5. Click on the **Export STL** button or press the *Enter* key:



Now, the STL file will be available to send to a 3D printer or printer service.

If multiple objects are selected, the exported STL will have the objects in the same relative orientation to each other; therefore, it's important that they don't overlap and are printable as oriented. It's often preferable to save separate objects in separate files.

Summary

A recent poll of online social sites about 3D printing showed Blender was the most popular choice for creating models for 3D printing. The reason is obvious, taking into account Blender's vast functionality. However, it could just as well be that more people were talking about Blender because of its challenging learning curve.

Blender is capable of creating simple primitive shapes, viewing them from any angle, transforming them with precision, and manipulating their individual vertices, edges, and faces in the powerful editing mode. The model can then be exported to a file, ready to be 3D printed.

Blender has many functions not even covered in this chapter, such as sculpting, skeletal manipulation, and how to use individual modifiers to achieve specific results. As these functions become important for individual projects, they will be covered.

Hopefully, this chapter served to introduce how powerful and comprehensive Blender is. However, Blender's comprehensive nature comes at the cost of being complex, which can be overwhelming. But don't worry. Keep this chapter at hand to act as a reference and a crutch in future projects until Blender's functions become second nature. It usually only takes one project before Blender becomes second nature. Once Blender is familiar, no other 3D modelling software will be necessary.

Chapter 3. The Octopus Pencil Holder

3D printing makes it easy to combine form and function. Why have just a pencil holder when you can have a pencil holder that looks like, say, a cute octopus? This project is an excellent starting project because it demonstrates a simple but versatile modeling technique that involves editing a simple mesh and smoothing it to create more detail.

This project will involve simple selection techniques unique to **Edit** mode, modification commands in **Edit** mode, and applying modifiers to soften and combine shapes. This technique alone can be used to make an unlimited number of cool things once mastered.

In this chapter, we will cover:

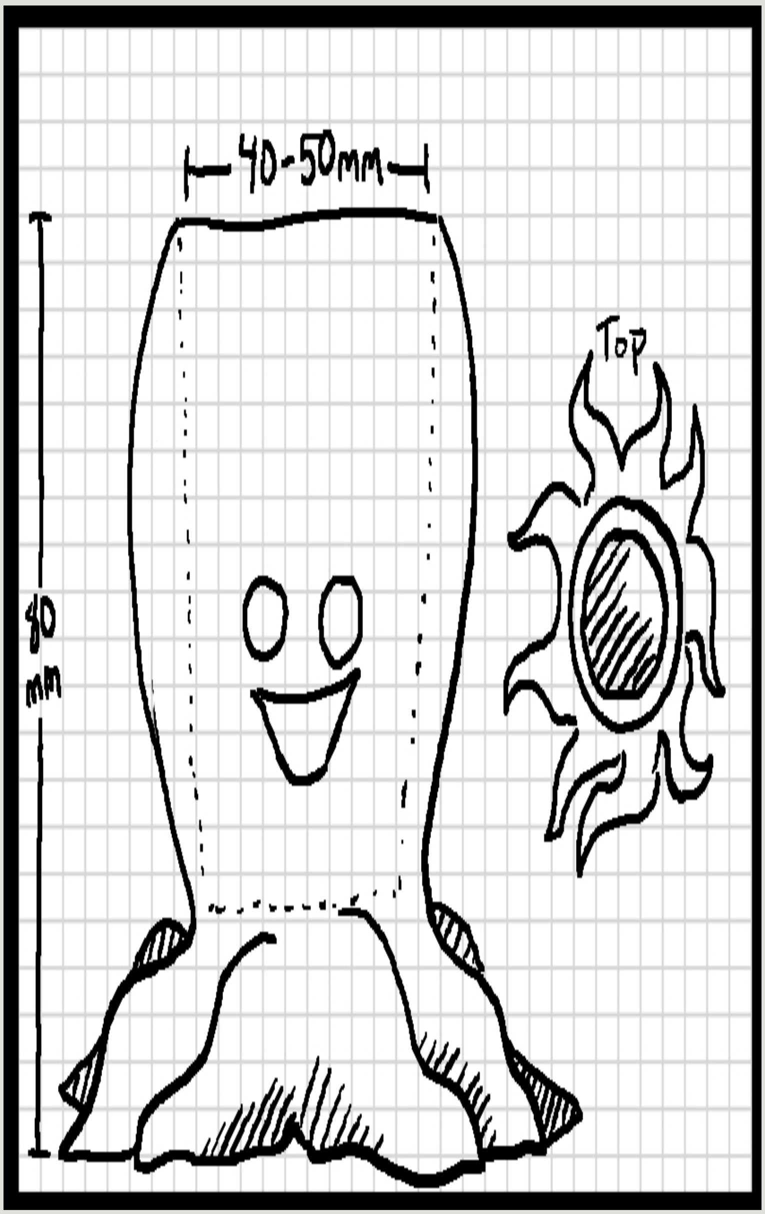
- Planning the project
- The first basic shape
- Smoothing the mesh with modifiers
- Bending the tentacles
- Flattening the bottom
- Renaming objects
- Adding a pencil cup
- Adding a face
- Finishing touches

Planning the project

A pencil holder is basically a cup with a sturdy base that can be used to hold objects taller than itself, such as pens, pencils, and other items that

might otherwise clutter up a desk. A cute, cartoony octopus's tentacles will provide the perfect base and prevent the cup from tipping over.

The hole in the cup should be roughly cylindrical, about 40-50 mm wide at the base. The whole thing should stand at least 80 mm tall. And, of course, it needs a cute and friendly face, something like this:

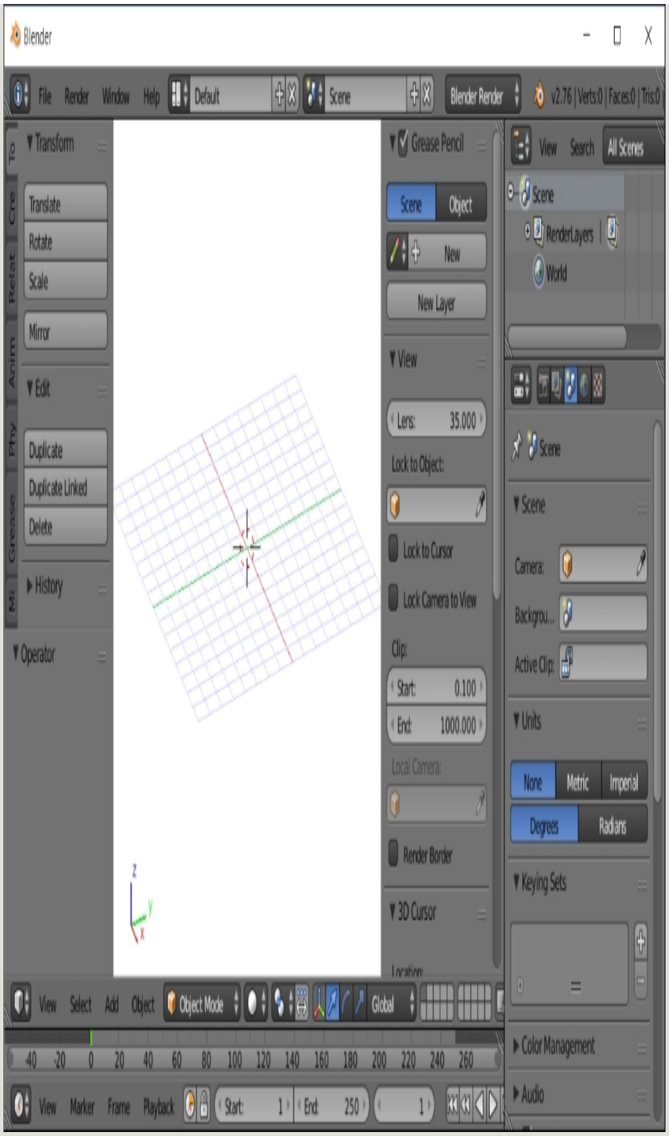


The first basic shape

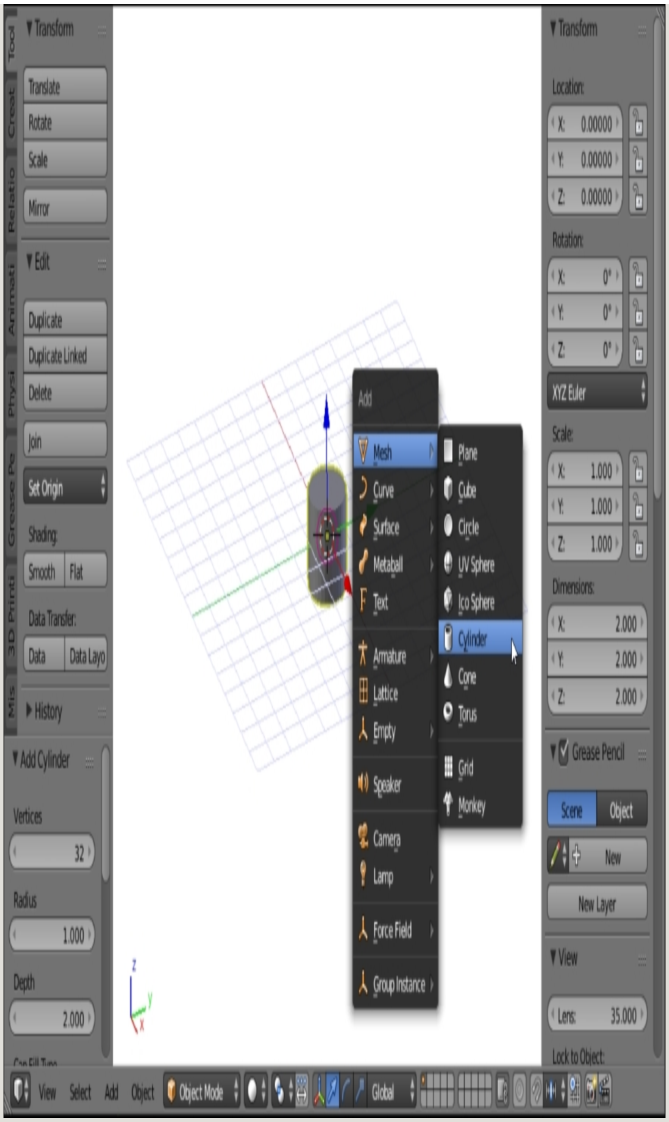
This project is going to take advantage of several powerful editing tools that Bender provides. The first one is going to be the `extrude` operator. Extruding takes its name from a process for making things in real life, but in 3D modelling, extruding takes a selected part of an existing model and creates new geometry on the edge of the selected parts, so that the original can be moved away but remain attached to where it came from. The result is a new shape that can then be edited.

Let's get ready; it's time to get to work:

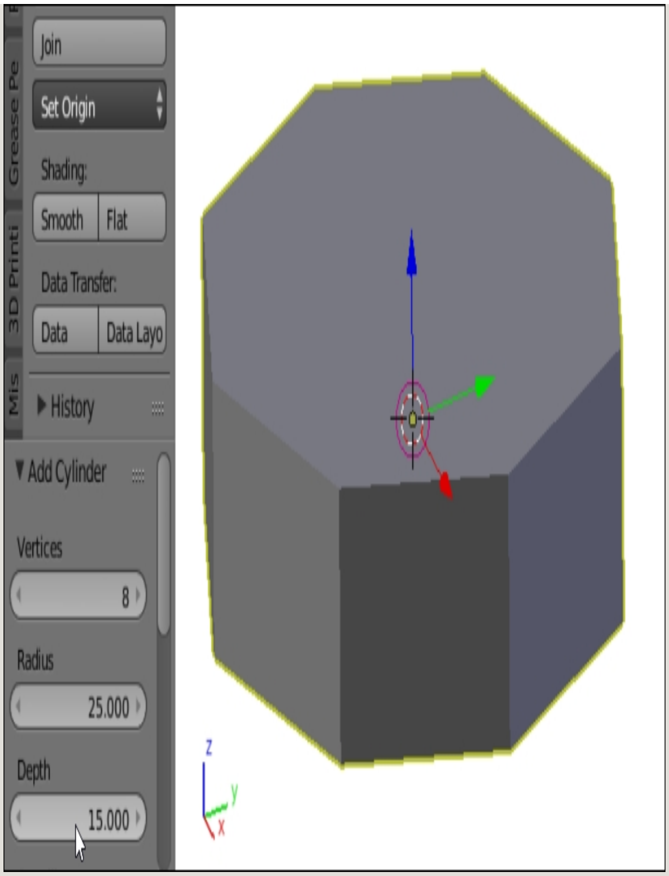
1. Open Blender, select all (*A*) in the default scene, and clear (*X*) it:



2. Next, add (*Shift* + *A*) a cylinder object to the scene:



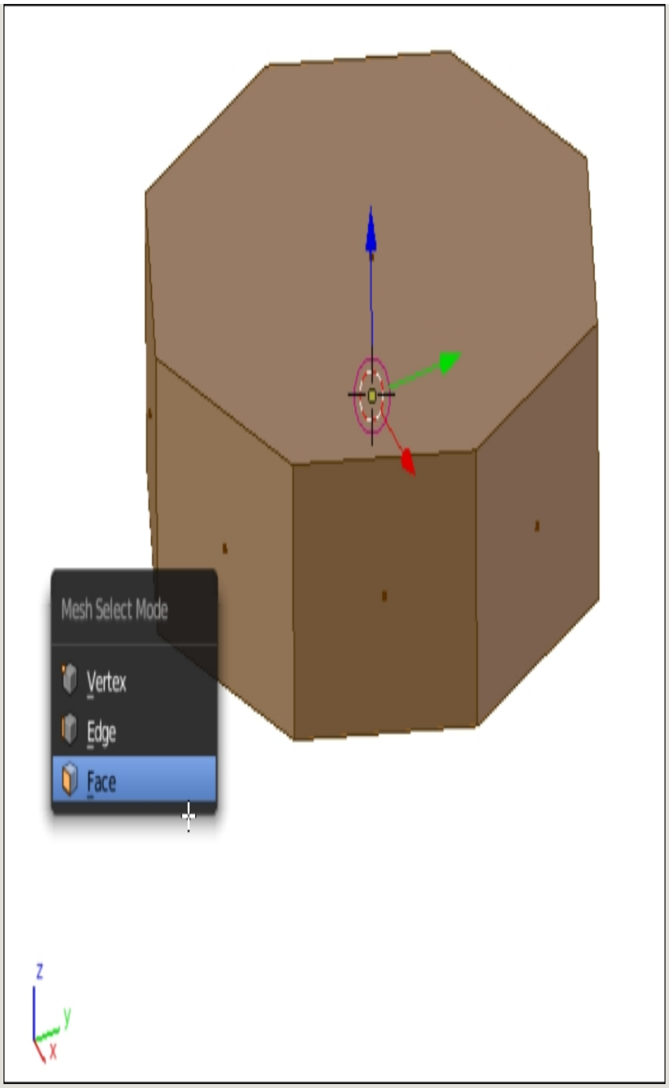
3. In the **Tool Shelf**, open the parameters for **Add Cylinder** and change the number of **Vertices** from 32 to 8. Change the **Radius** field to 25 and the depth to 15:



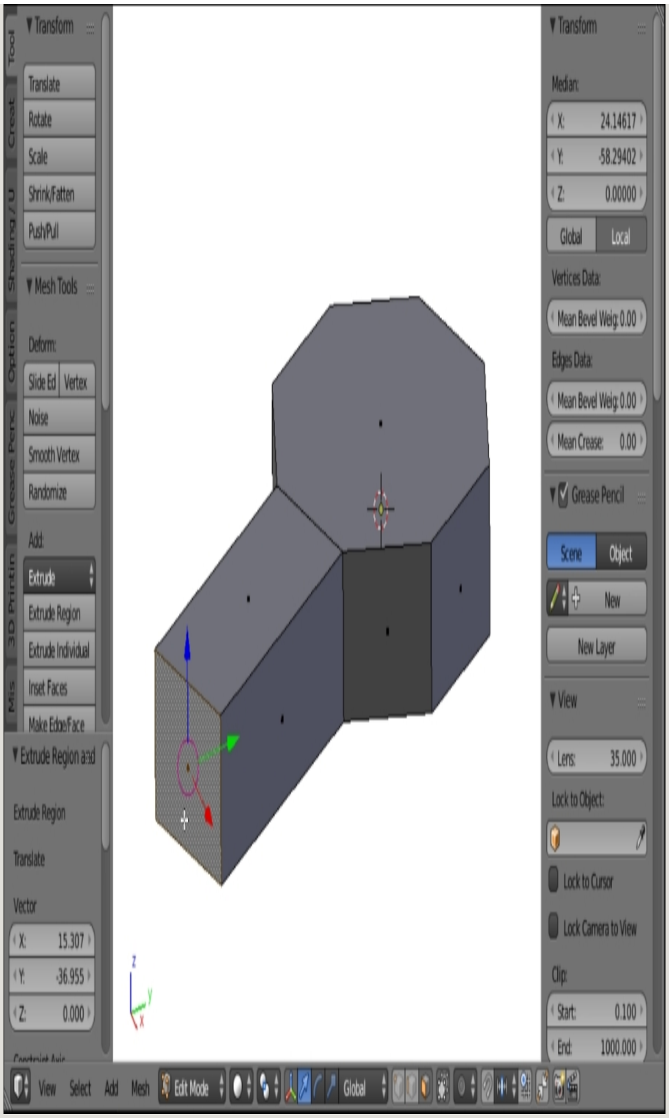
NOTE

Remember that these settings are only available when an object is first created. Selecting anything else, performing a transformation or action, or even deselecting the object will make these setting disappear. If the settings are gone, the only way to get them back is to delete and re-add the object to the scene.

4. Enter **Edit** mode (*Tab*) and switch to **Face Select** mode (*Ctrl* +*Tab*):



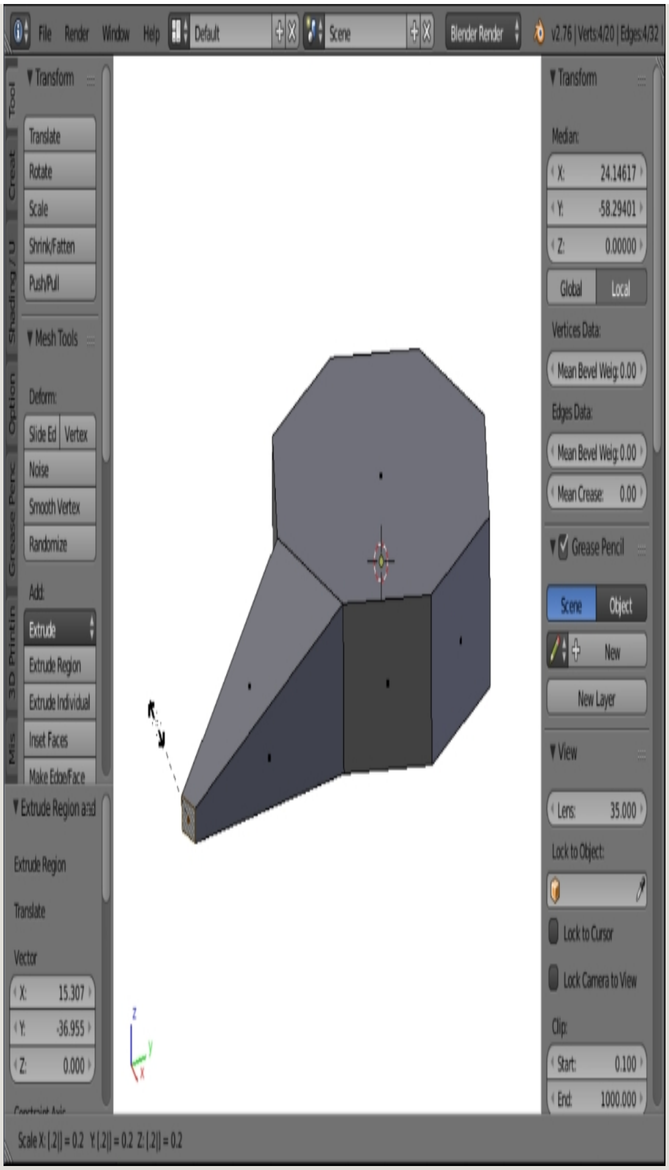
5. Deselect all faces (*A*).
6. Select one of the vertical sides of the cylinder.
7. In the **3D View** menu, navigate to **Mesh** | **Extrude** | **Region**, or press *E* on the keyboard.
8. Move the mouse until the leg is long enough for you to just type `40` to extrude it exactly `40` mm.
9. Press *Enter* or click on the select mouse button to complete the action:



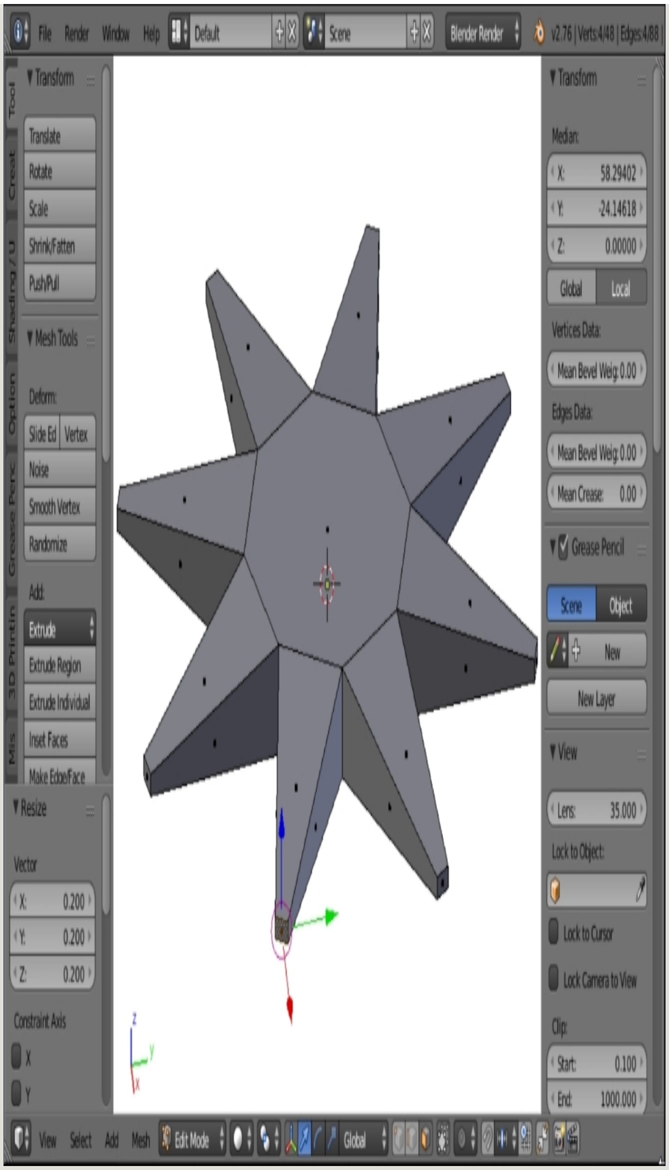
NOTE

Like all commands, the `extrude` command can be cancelled by pressing *Esc* key or clicking the non-select mouse button before the action is completed.

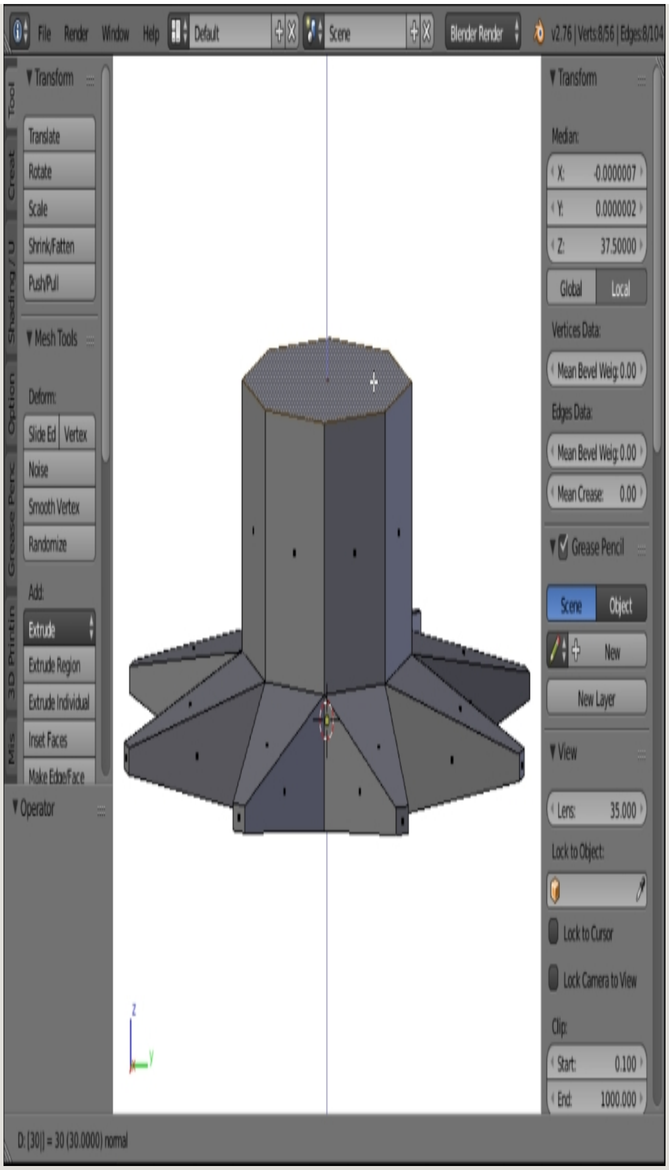
10. To give this arm a gentle, tentacle-like shape, scale the selected face at the end of the arm (*S*) down to about 20 percent (*.* + *2*):



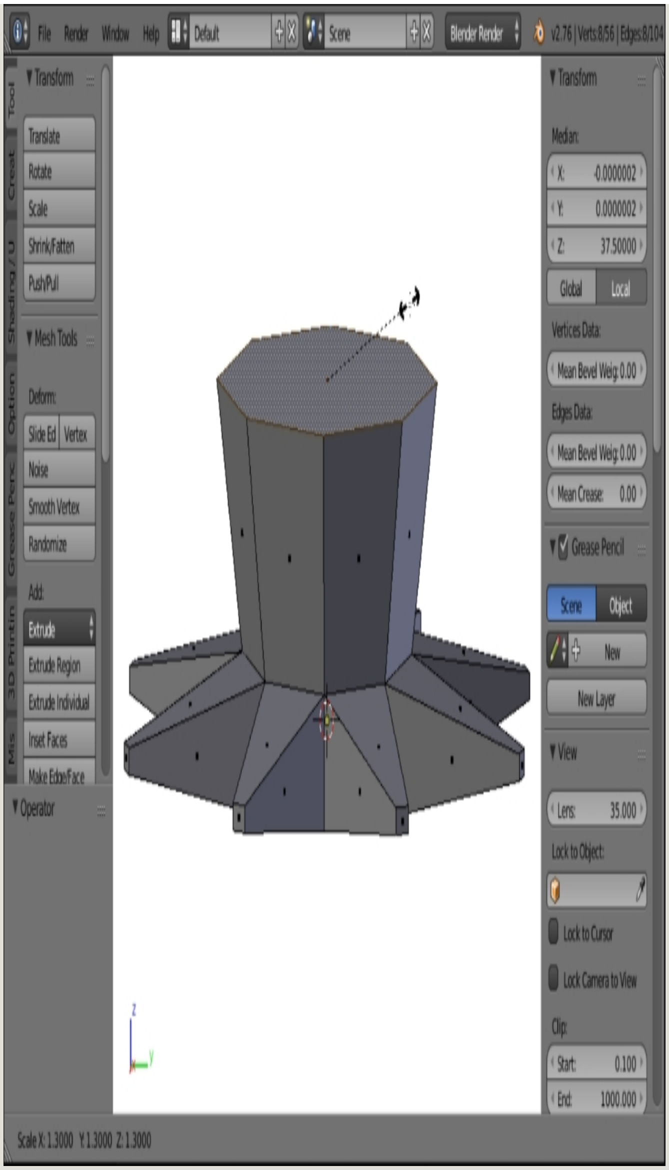
11. Repeat the extruding and scaling process with the other seven vertical faces of the cylinder to make all eight tentacles:



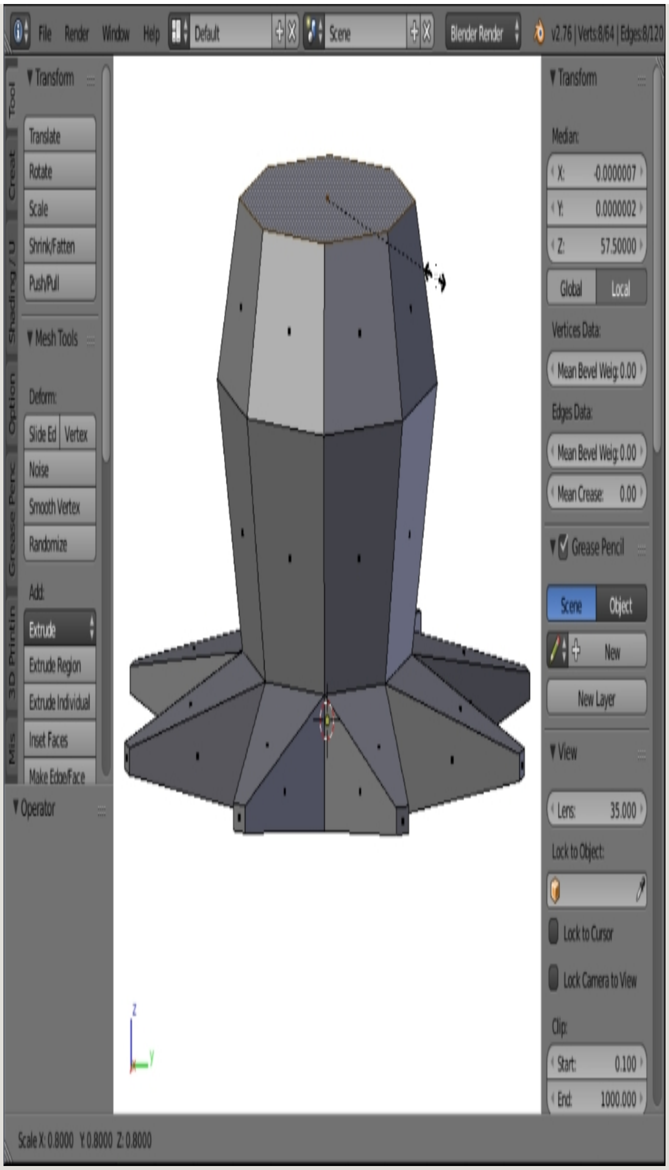
12. Select the top face of the cylinder and extrude (*E*) it about 30 mm:



13. Then scale (*S*) it up just a little bit to make the head bulbous:



14. Extrude (*E*) the top again, this time about 20 mm, and scale (*S*) it in to give the top a more rounded shape:

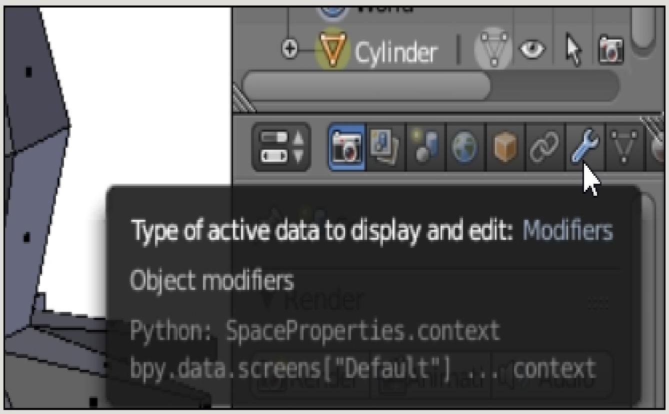


Like making a carving out of stone, these first stages are rough, but the model will be refined later. As it is, this basic shape has got a head and eight arms. That's enough like an octopus to get started.

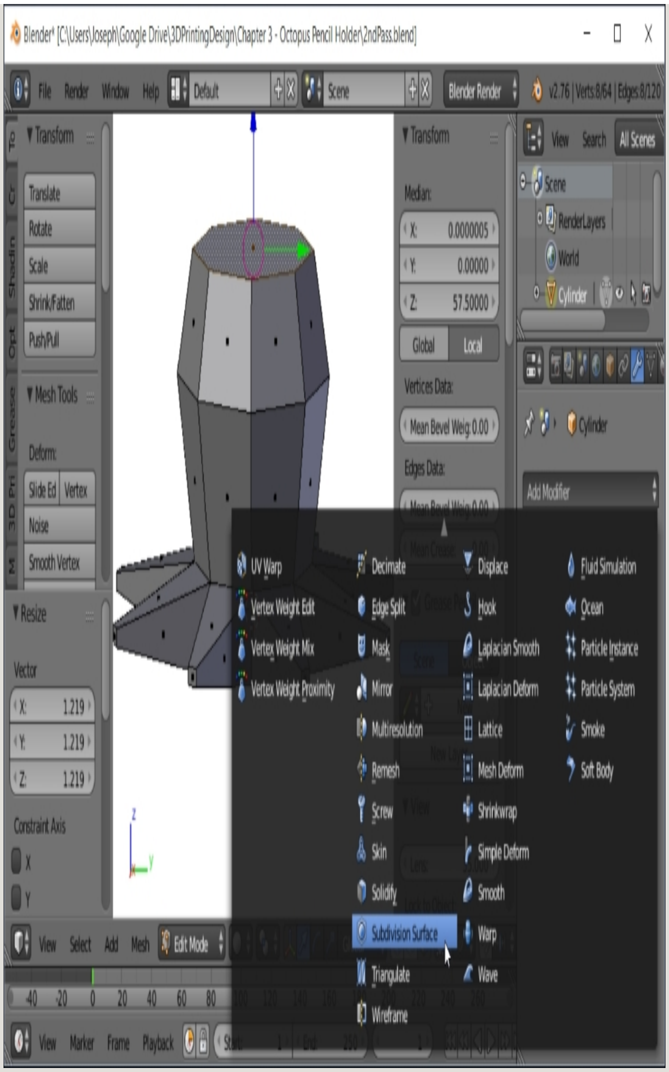
Smoothing the mesh with modifiers

This blocky octopus is fine for editing; it doesn't have many parts to keep track of. But it needs to be much smoother for the final result. Fortunately, there's a way to increase the smoothness of the model while retaining a simple geometry that is easy to edit. This is done by adding a **Subdivision Surface** modifier to the object.

1. To add a **Subdivision Surface** modifier, click on the **Modifiers** tab in the **Properties** panel (the one that looks like a wrench):

2. Click on the **Add Modifier** button, and in the menu that appears, choose **Subdivision Surface** from the list:

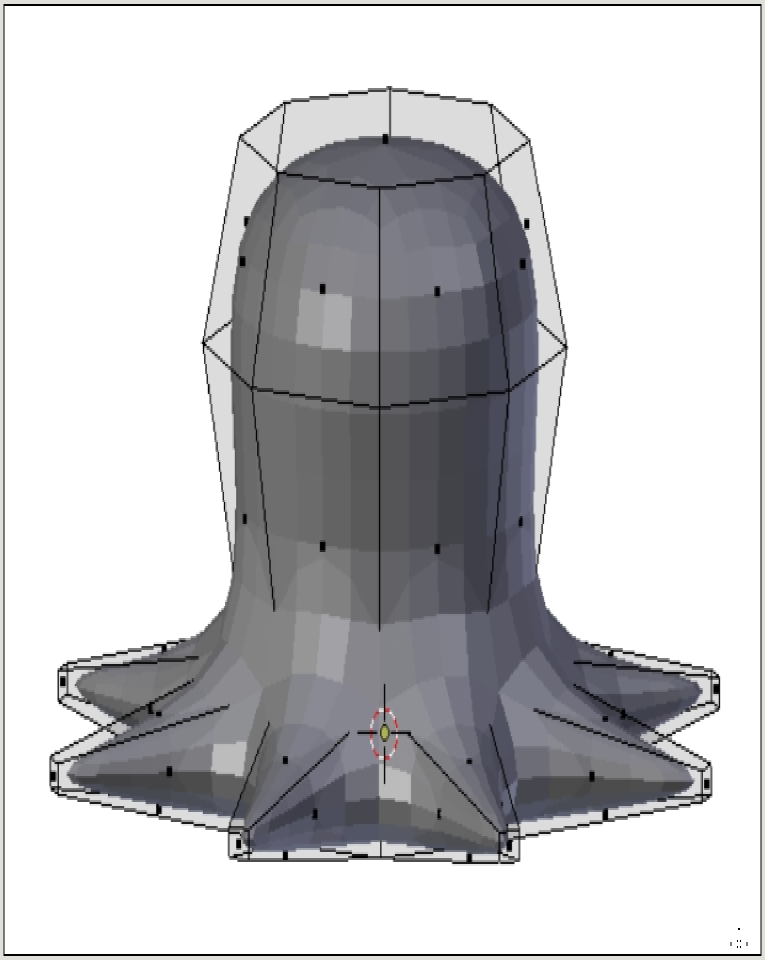


With the **Subdivision Surface** modifier on, the shape looks much smoother. While in **Edit** mode, it's clear that the original geometry is still there and acts as a sort of cage that defines the shape of the smoothed mesh. As long as the modifier isn't applied, the simpler geometry can be retained for editing.

TIP

Changing the **View** setting in the modifier will affect how smooth the mesh will be. Higher values will create more complex geometry that will look smoother but will slow down the computer more. There comes a point where increasing this setting won't make any noticeable difference. Generally, it's best to keep this setting high enough to have a good effect on the shape but low enough that it doesn't slow down the computer.

For now, set the **View** value to 2. This will give you something like this:



This isn't the final form, but it will get better. If seeing the smoothed mesh while working on the simpler mesh is confusing, the modifier can be temporarily turned off by clicking on its eyeball icon.

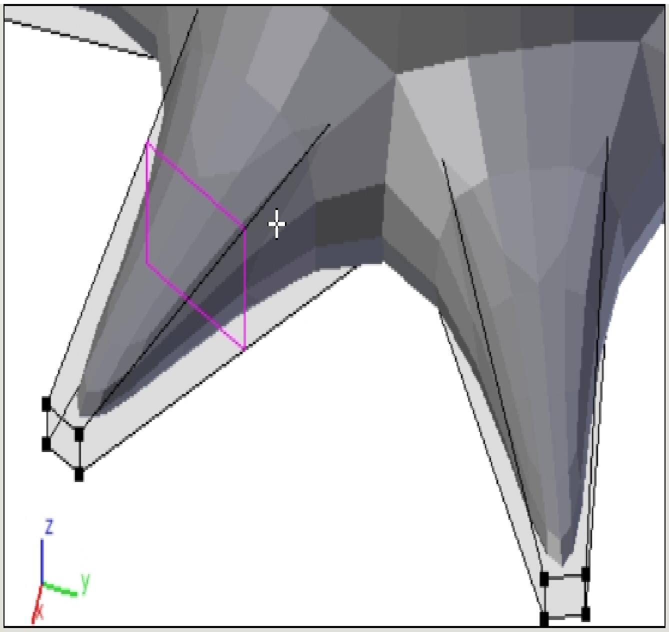
Bending the tentacles

Time to begin adding some details to the tentacles and give the model some personality.

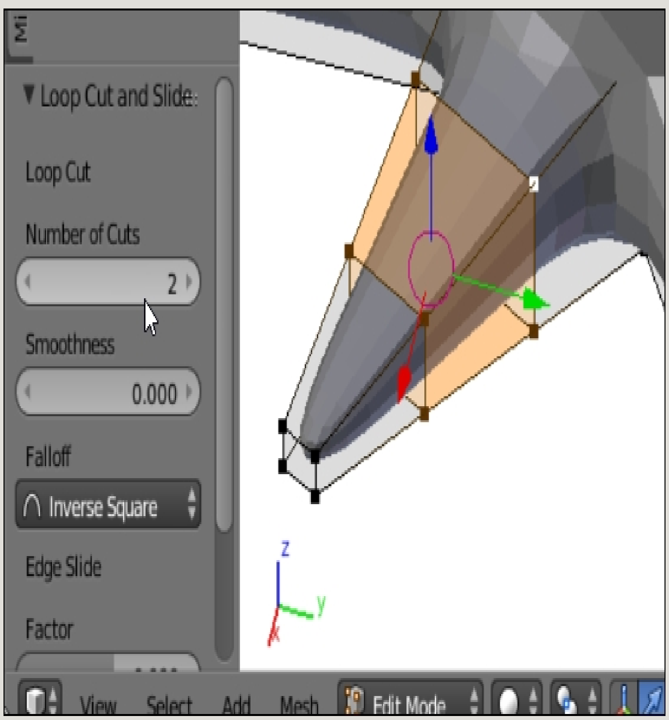
Before moving ahead, let's take a look at another powerful tool for editing meshes called loop cut or loop subdivide, which will be used when bending the tentacles. Loop cut adds points in the middle of an edge and all around a portion of the geometry. These points can then be transformed. Loop cut only works with edge or **Vertex Select** mode and will automatically switch modes when being used.

For our project, we'll be using loop cuts to bend the tentacles. Follow these steps:

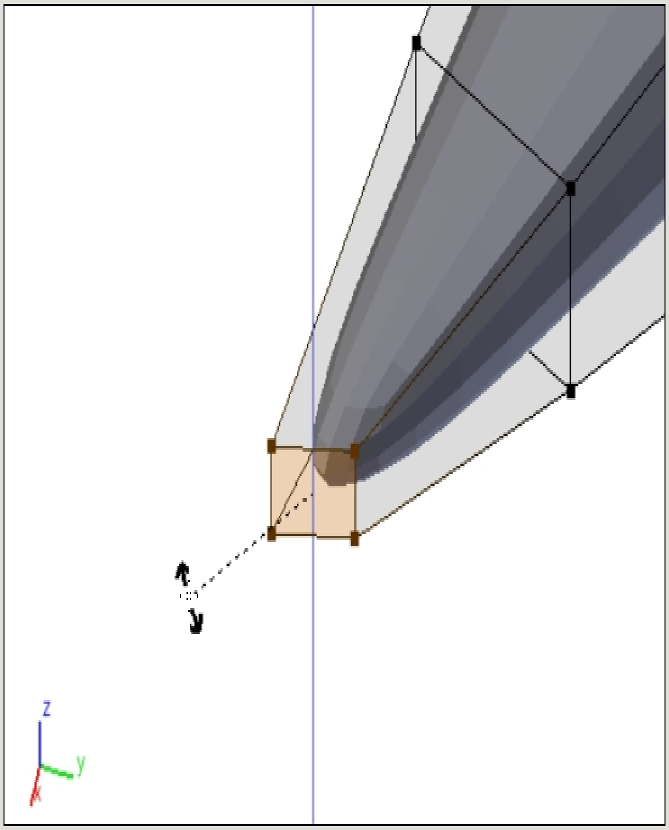
1. While still in **Edit** mode, switch to **Vertex Edit** mode (*Ctrl* + *Tab*).
2. Begin the loop cut operation by selecting **Loop Subdivide** under **Mesh** | **Edges** from the **3D View** menu or pressing *Ctrl* + *R*.
3. Move the mouse over the edge of one of the tentacles. Notice the preview line showing where the loop cut will be:



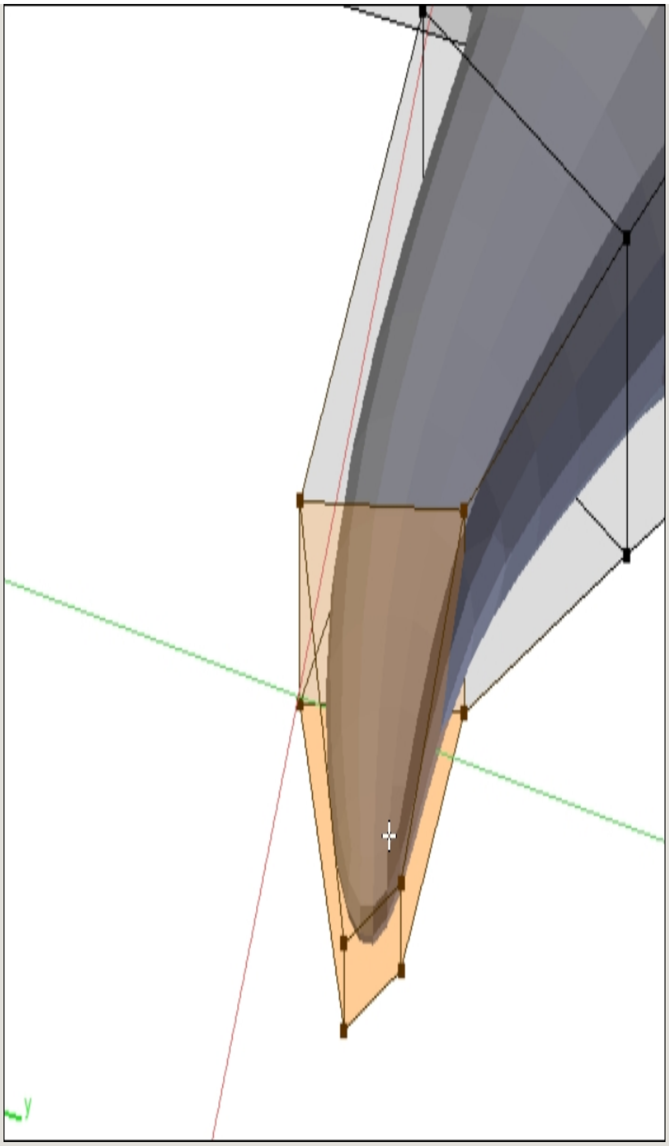
4. When the loop cut is where desired, click the select mouse button or press *Enter*. At this point, the loop isn't set and can still be slid back and forth by moving the mouse.
5. Try to keep this loop cut near the middle of the tentacle.
6. When it is where desired, press *Enter* or the select mouse button again to finish the operation.
7. After setting the loop cut, change the operator settings at the bottom of the **Tool Box** and change **Number of Cuts** to 2:



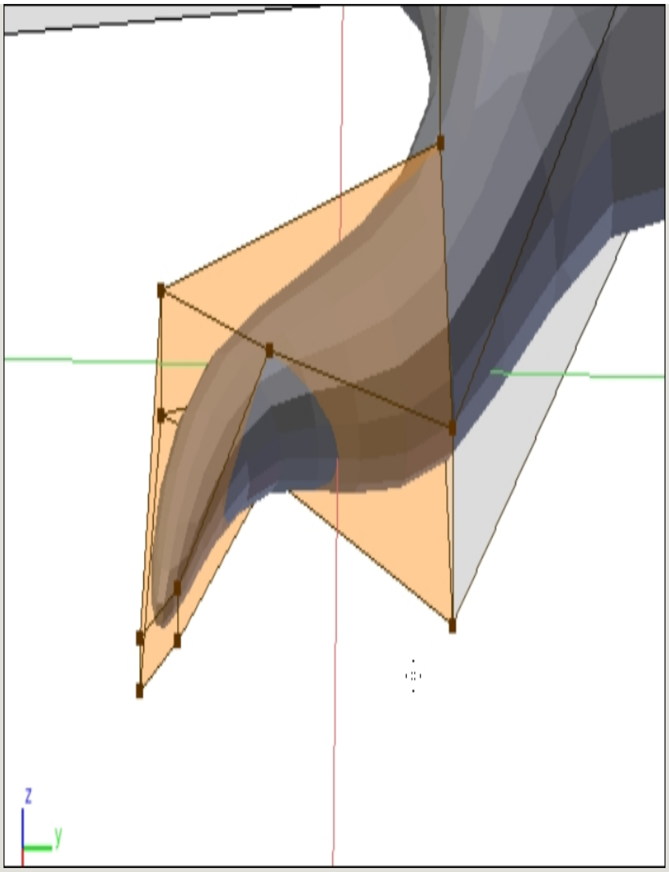
8. Deselect all points (*A*).
9. Select the points at the tip of the tentacle using either box (*B*) or circle (*C*) select.
10. Rotate (*R*) the points around the Z axis (*Z*).
11. Move (*G*) them along the X and Y axes (*Shift* + *Z*) to bend the tentacle a bit:



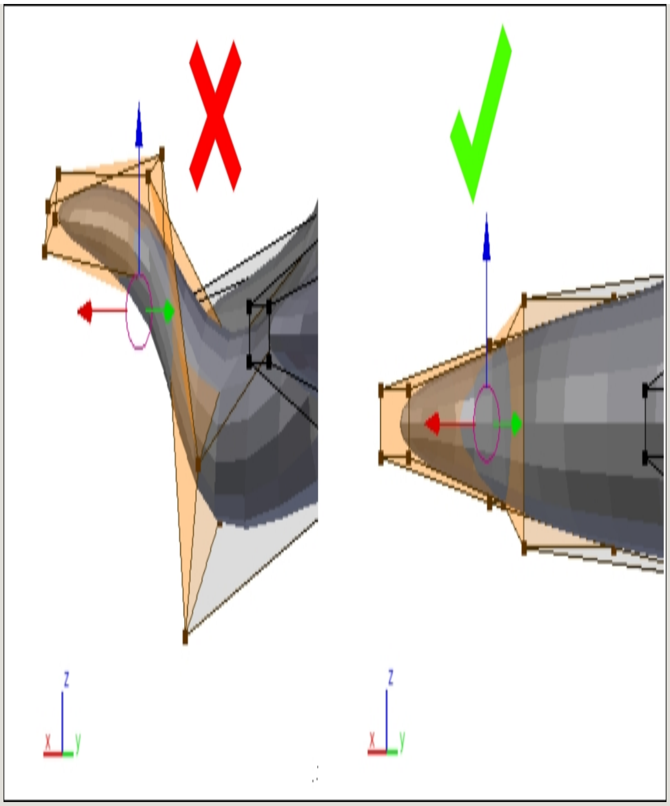
12. Expand the selection (*Ctrl* + Numpad *+*).
13. Again, rotate (*R*) the selection around the Z axis (*Z*) and move (*G*) it along the X and Y axes (*Shift* + *Z*) to bend the tentacle a bit more:



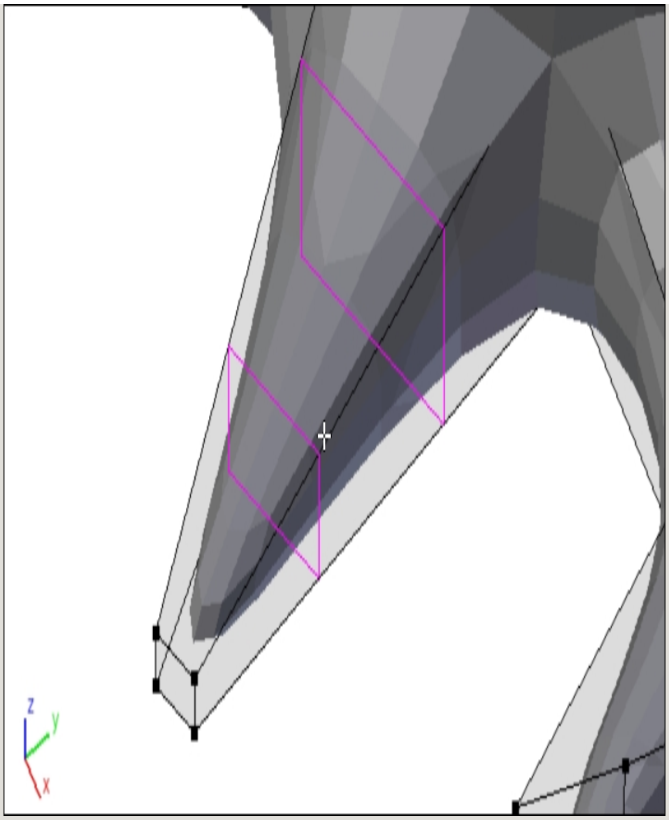
14. Expand the selection again. Rotate (*R*) and move (*G*) the tentacle, being careful to constrain the movement as before:



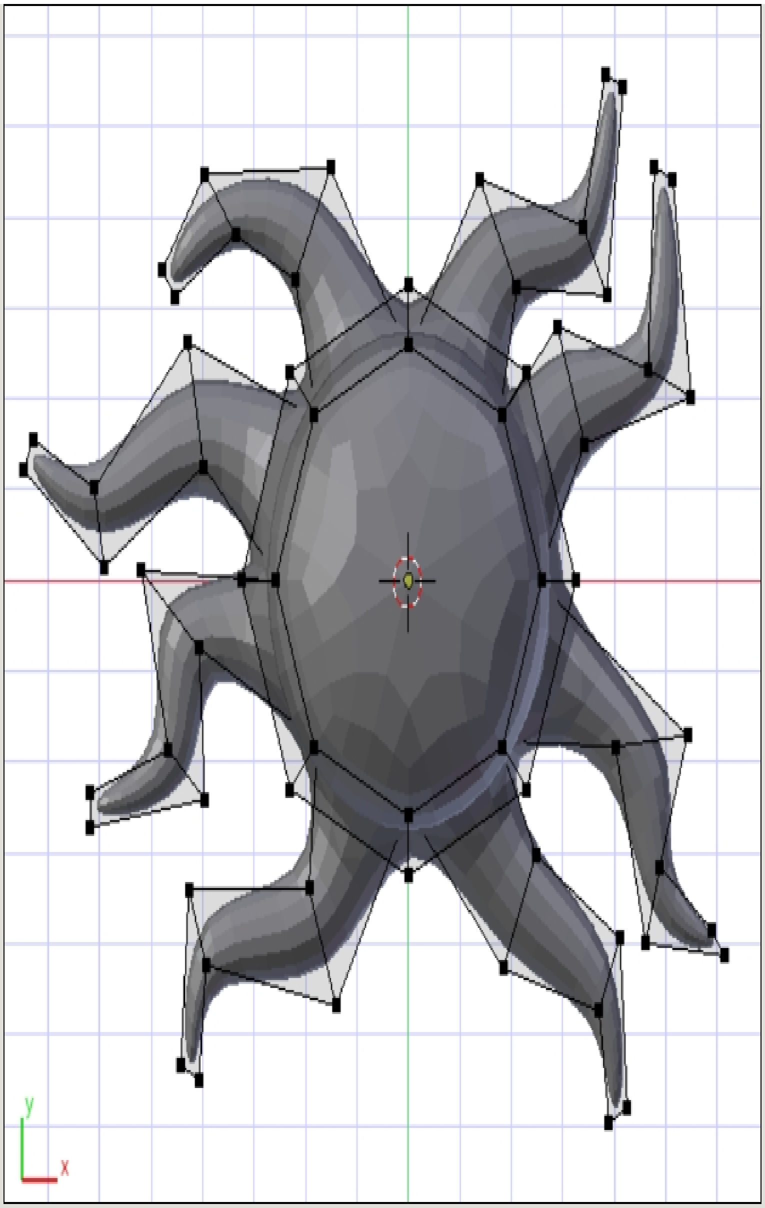
15. Adjust your view to check your tentacle. Make sure that from the front or side views, the twisty tentacle is still on the same level as the other tentacles and not twisted up and down. If it is twisted, then your movements weren't constrained properly:



16. If it is not correct, then undo (*Ctrl* + *Z*) your movements and try again.
17. Another way to ensure that the movements are constrained properly is by performing all your transformations while in the top view (Numpad *7*). The trick to this method is to select points in **Wireframe** view (Z) to get all the points, not just the ones on top.
18. Move to the next tentacle and loop cut it. This time, while the loop location is being display, before clicking the mouse or pressing *Enter* the first time, try pressing the *2* key to quickly change the number of cuts to 2. This is a shortcut method that can optionally be used instead of changing the parameters after the cut:



Work around the body, cutting, selecting, moving, and rotating each tentacle. Give each one a different twist, being careful not to overlap them. If the tentacles overlap, the model won't print properly when exported.



Your octopus doesn't need to look exactly like this. Make it your own. When all the tentacles have been given details, it should look much more like an octopus.

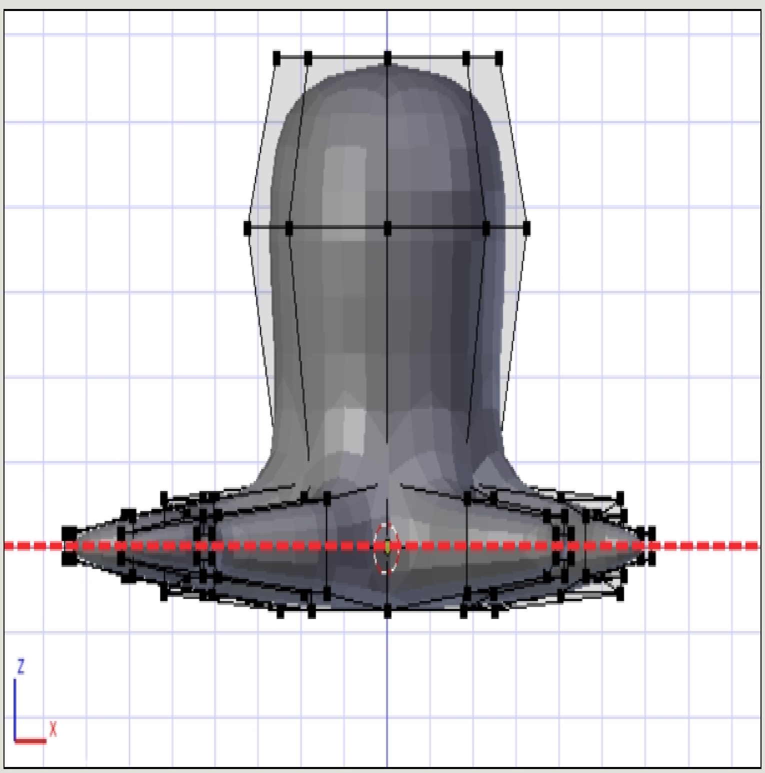
TIP

Some 3D printers won't be able to print this model if it's too big, so keep those tentacles tucked in and don't let them fly too far. The smaller 3D printers' build areas are about 150 mm or 6 inches squared. If the total area of your model is more than this, you'll need to think about the printer you're designing this for.

Flattening the bottom

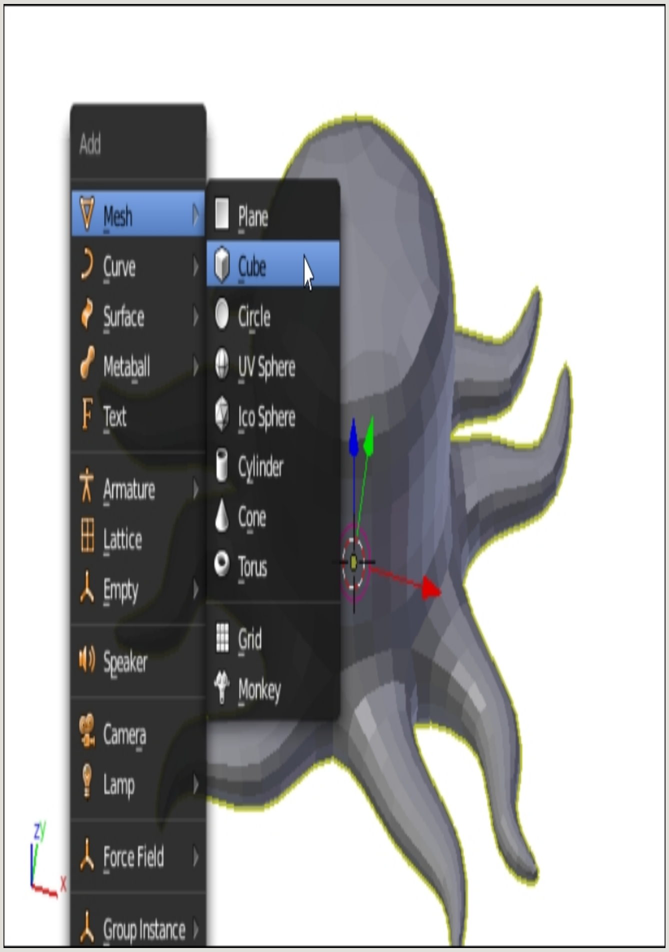
An FFF-friendly 3D model needs to have a solid, flat base. There is more than one way the bottom of a model can be flattened. For one, the geometry can be edited to be flat. Alternatively, a floor object can be created and cut out of the object using a `Boolean` modifier. Since this project is using the **Subdivision Surface** modifier, editing the geometry can be complicated, so the floor method will be used.

Looking at the model from the front **Ortho** view (Numpad *1* and Numpad *5*), it's clear that the parts of the tentacles that sit below the `X` and `Y` axes (the red or green line, depending on the view) is where the model needs to be cut off to make a flat, printable base:

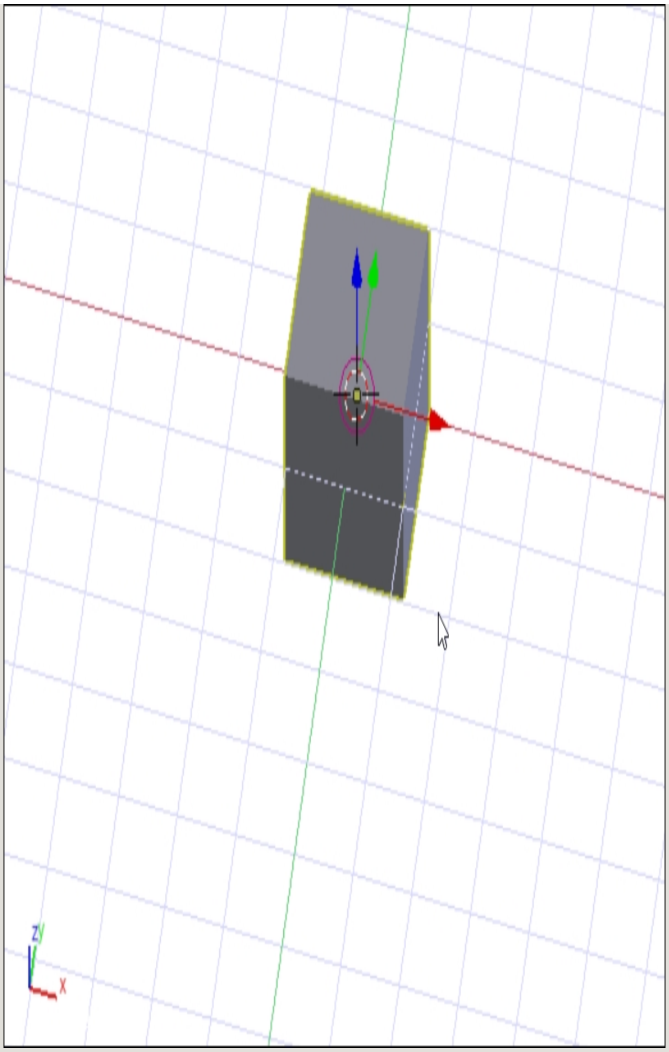


So, let's get started:

1. First, exit **Edit** mode (*Tab*).
2. Be sure the 3D cursor is at the **3D View**'s origin point (*Shift* + *C*). Then create a cube (*Shift* + *A*) by navigating to **Mesh** | **Cube**:

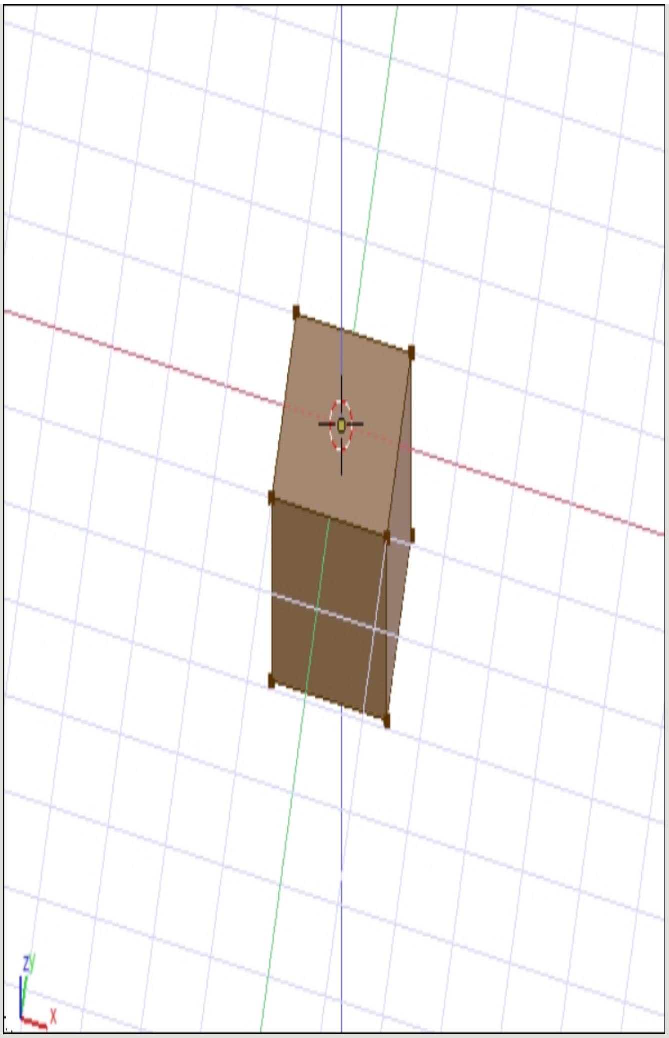
   

   The cube cannot be seen in solid view because it's completely inside the octopus, so with the cube still selected, switch to local view (Numpad slash */*) to view the cube alone:

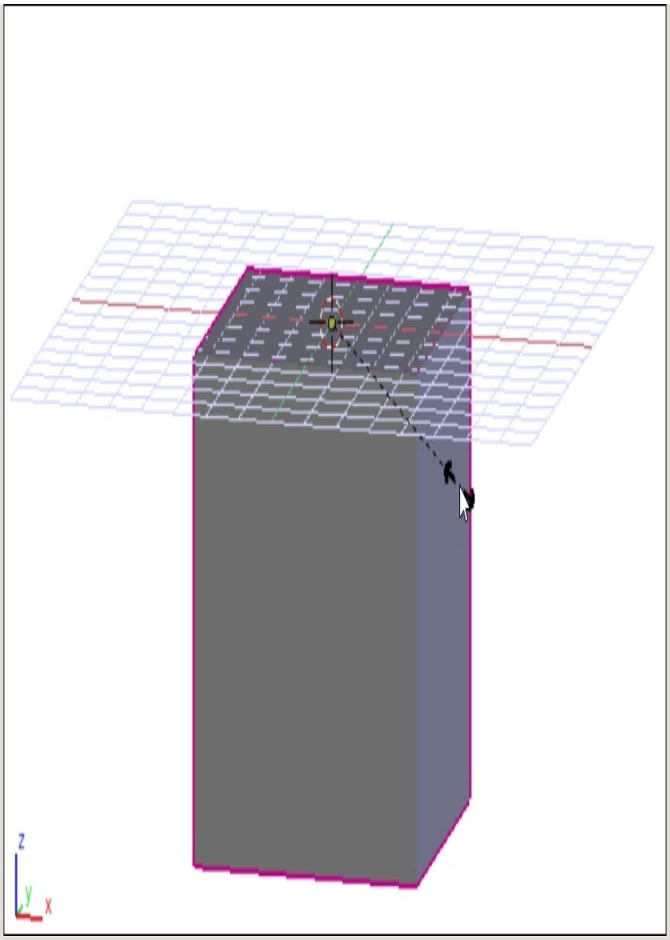


The cube is half above and half below the origin. Fortunately, there's an easy way to make a floor and ensure that, no matter how, it's scaled so that the top remains on the XY plane.

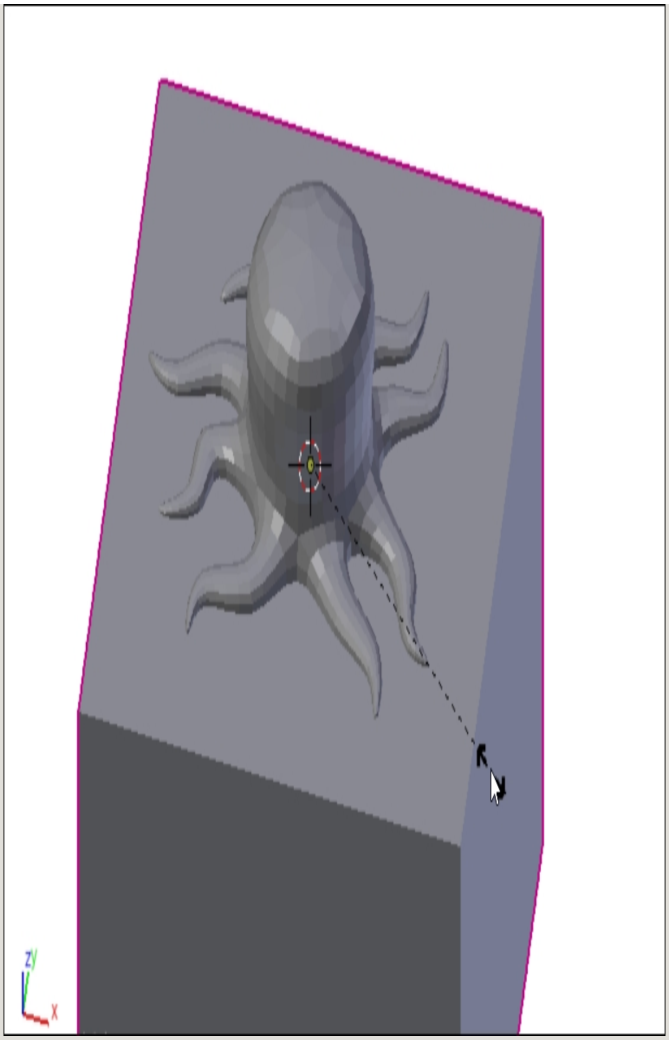
3. In **Edit** mode (*Tab*), with all points selected (which should be by default; *A* is the hotkey if it's not), move (*G*) them along the Z axis (*Z*) 1 unit (- + *1*):



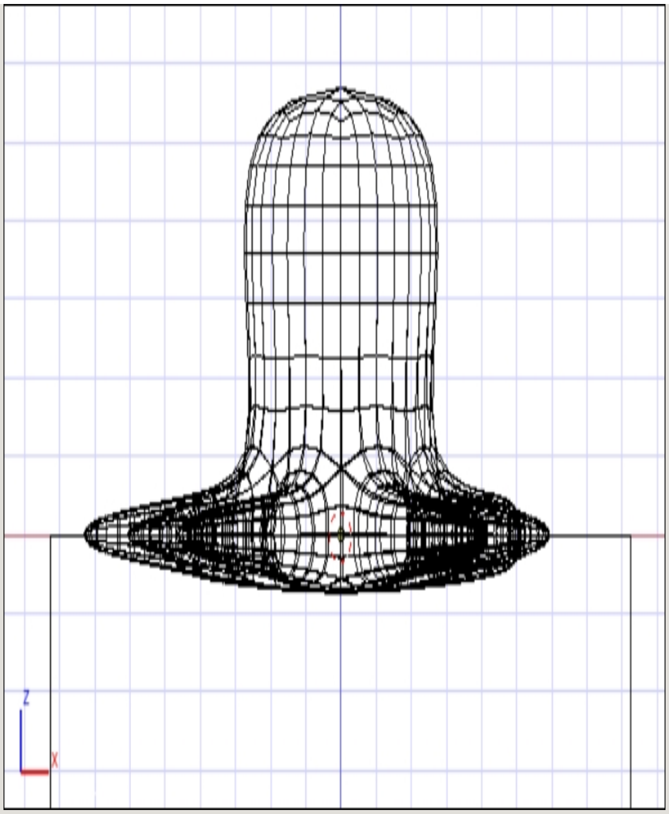
4. Then exit **Edit** mode (*Tab*) and scale (*S*) the cube. Notice that the top remains on the XY plane. This is because object transformations are made in relation to the object's origin. When the points were all moved in **Edit** mode, the origin wasn't affected. So now when scaling the top, because it's in line with the cube's origin, it remains on the same plane:



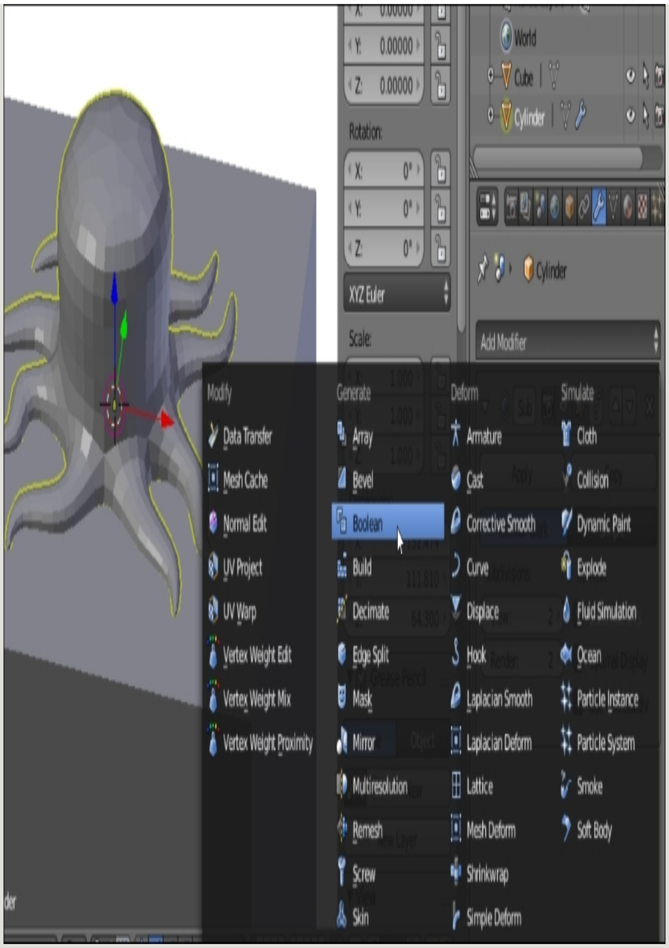
5. Exit **Local** view (Numpad slash */*) and scale (*S*) the cube until it covers the bottom of the octopus's body:



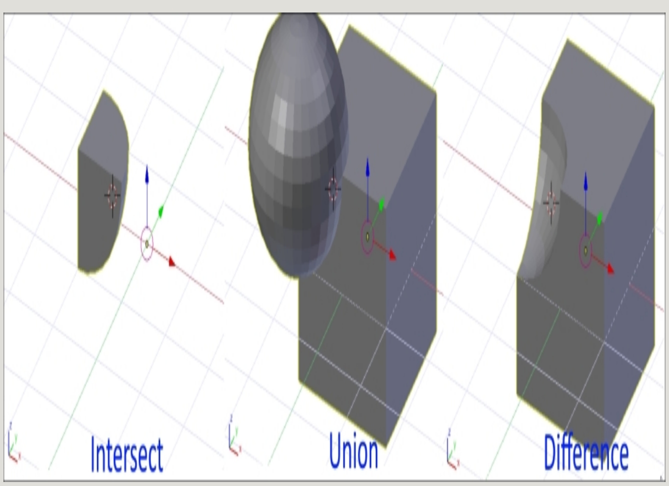
In **Solid** view (*Z*), it may look as if the octopus has a flat bottom, but in **Wireframe** view (*Z*), it is clear that the cube is only hiding the bottom part. For the final model, the bottom needs to be actually flat:



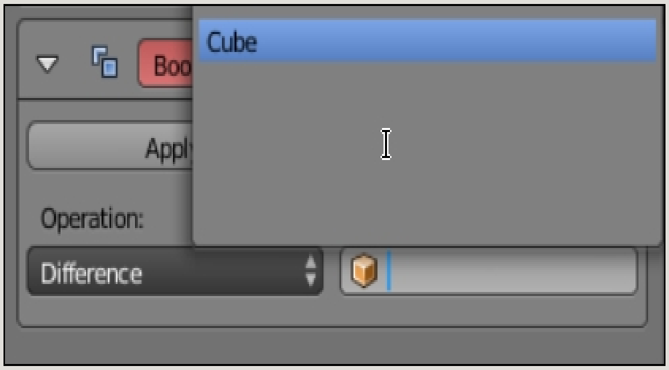
6. Select the octopus body.
7. In the **Modifier** tab, add a **Boolean** modifier to the octopus body:



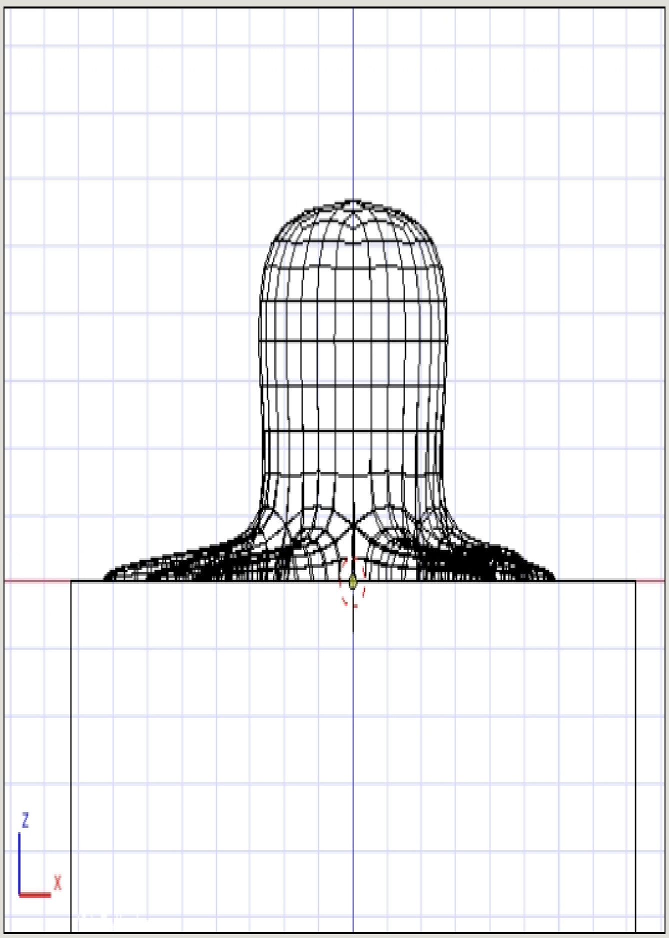
8. The **Boolean** modifier has options to combine the selected object with another object in a number of ways: **Intersect**, **Union**, and **Difference**. **Union** joins the two objects so that they become one, **Difference** cuts the second object out of the first, and **Intersect** leaves only the part where the two objects overlap. **Boolean** is a powerful tool and it's good to be aware of how to use it:



9. Change the **Operation** value of the **Boolean** modifier to **Difference**. Click on the **Object** text field and choose **Cube** from the list that pops up:



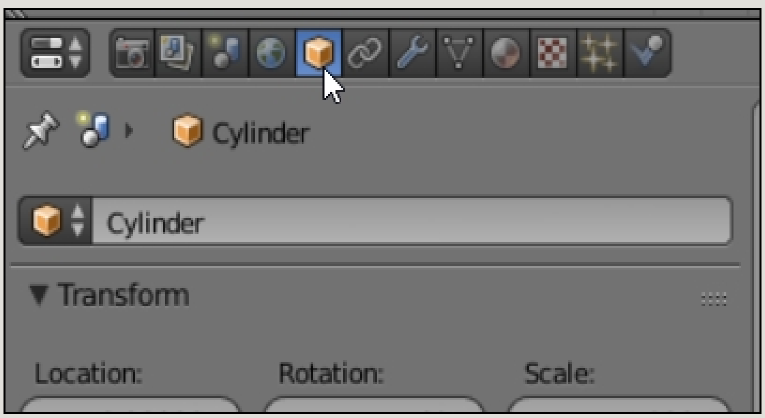
10. Now, the octopus body actually has a flat bottom suitable for 3D printing. This can be confirmed in **Wireframe** view (*Z*):



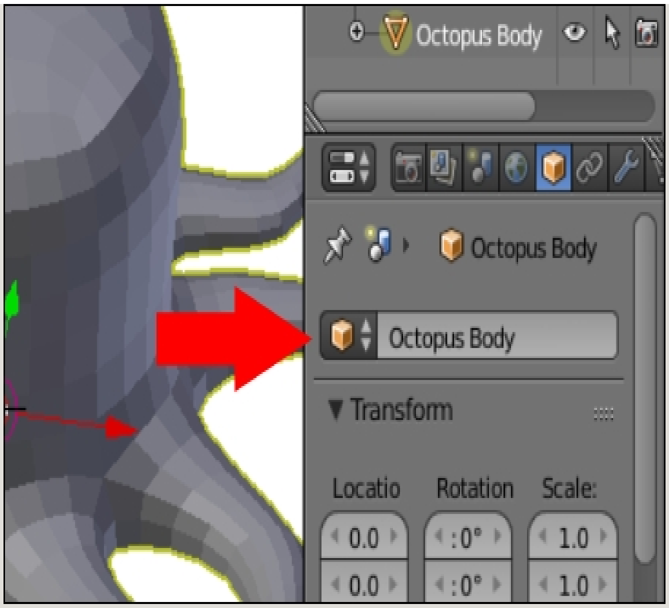
Renaming objects

Now that there is more than one object in the scene, leaving their names as the basic shapes they started as can be confusing, especially as other objects enter the scene. It's best practice to name objects as something more descriptive. Let's look at how to rename an object.

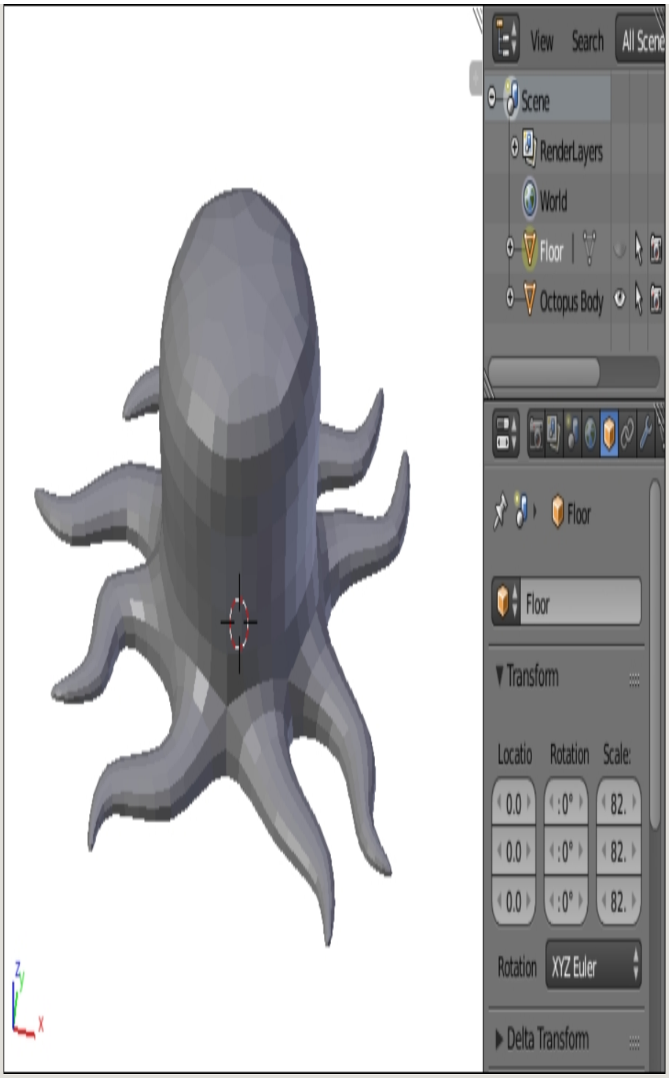
Objects can be renamed in the **Object** tab in the properties panel—the one marked with an orange cube icon:



1. Select the octopus body in the **3D View** (or **Cylinder** in the outliner panel). In the **Objects** menu, click on the name, currently **Cylinder**, and change it to **Octopus Body**:



2. Now, select the cube, and in the **Objects** menu, change its name to **Floor**.
3. Since the floor has served its purpose, in order to avoid it getting in the way or getting accidently transformed, hide it from view by selecting it and then navigating to **Object** | **Show/Hide** | **Hide** in the **3D View** menu or by pressing *H*. The object is still in the outliner view but hidden in the **3D View**:

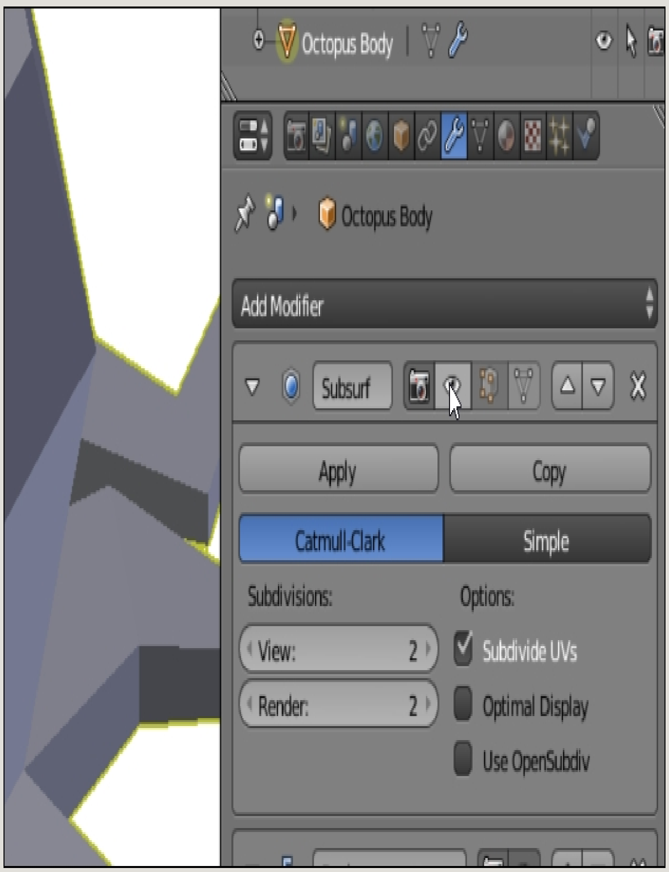


Hidden objects can be unhidden by going to **Object** | **Show/Hide** | **Show All**, pressing *Alt* + *H*, or by pressing the eye icon next to the object in the **Outliner** panel.

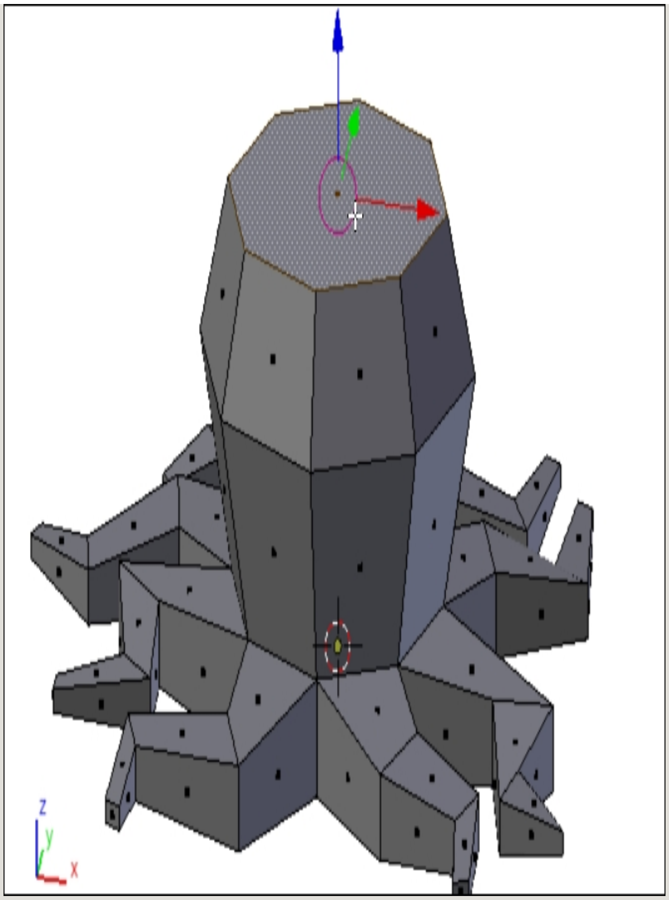
Adding a pencil cup

The octopus model so far is appealing, but it can also be made functional. The plan for this project was a cup holder, so it needs to have the shape changed so that pencils can be put inside it.

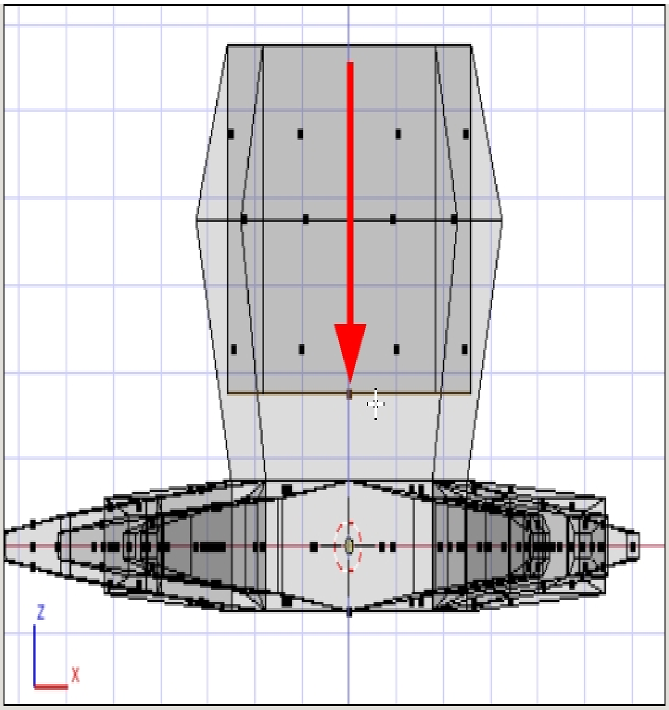
1. To start, temporarily turn off the **Subdivision Surface** modifier by locating the **Subsurf** modifier in the **Modifiers** tab and clicking on the eye icon for it. Now the simplified geometry is easier to work with:

2. Go to **Face Select** (*Ctrl* + *Tab*) mode in **Edit** mode (*Tab*). Now, select the topmost face of the octopus:

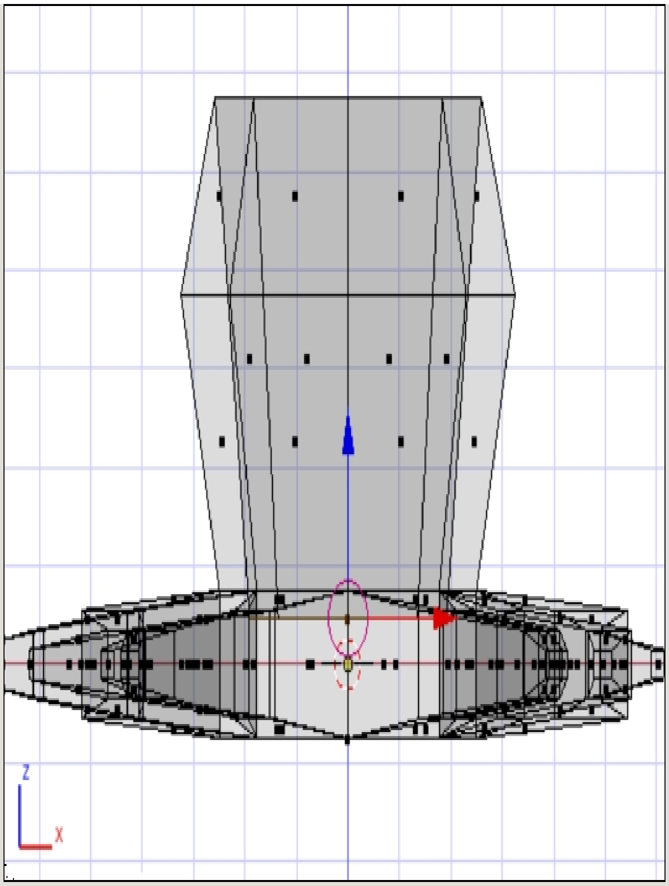


3. Switch to **Wireframe** (*Z*) and front **Ortho** view (Numpad *1*) and **Extrude** (*E*) the top face down into the body. Stop just a little above the red X axis line:

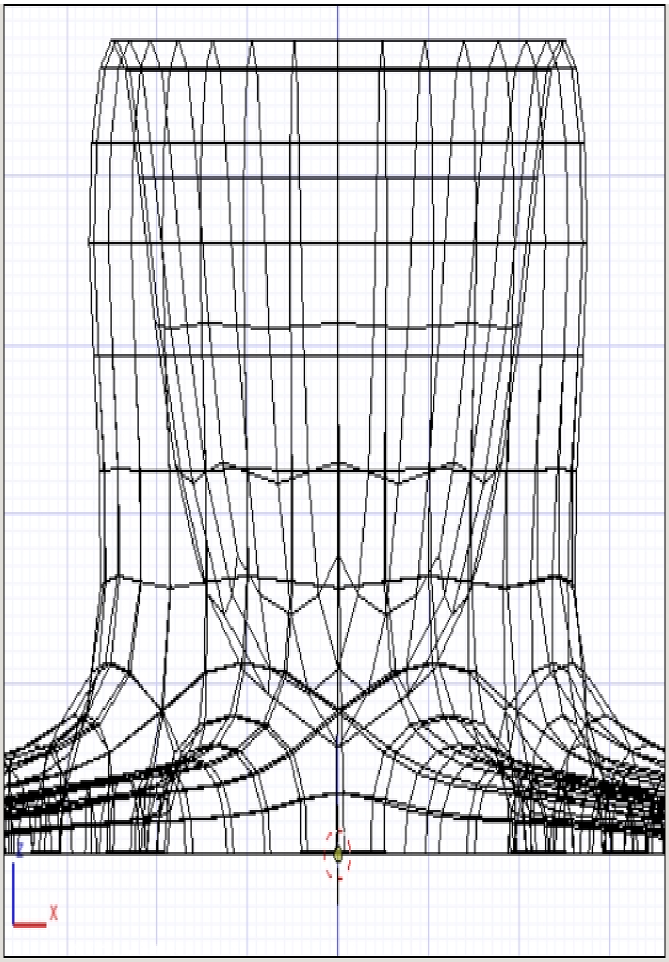


NOTE

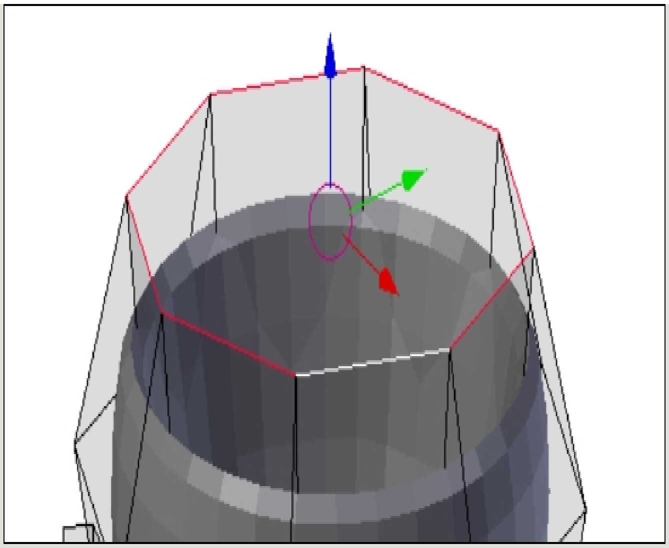
It may be necessary to scale (*S*) the bottom of the cup a little so that it fits inside the body.



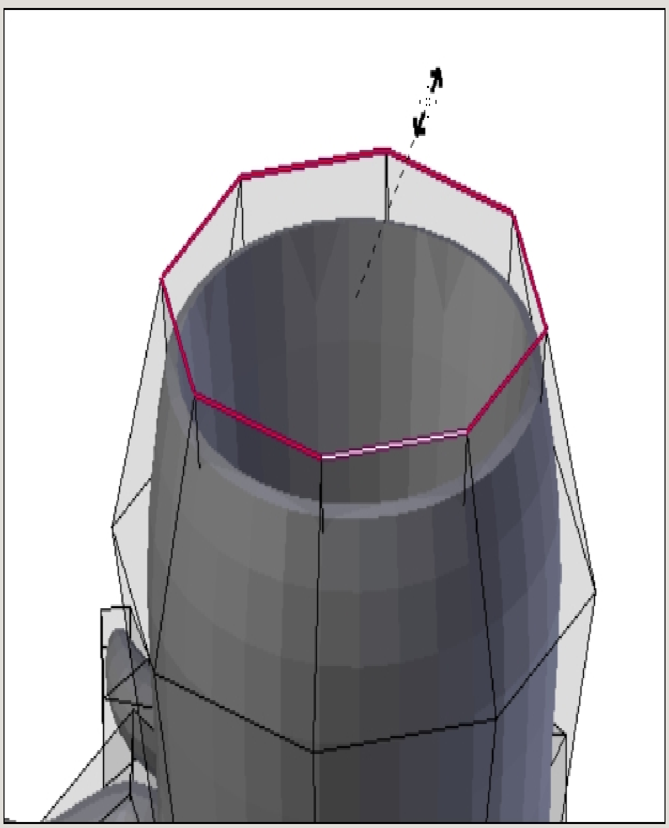
4. Now the shape is generally right for a pencil holder. Turn the **Subdivision Surface** modifier back on by clicking on its eye icon in the **Modifiers** tab. You'll notice that the cup bottom is a bit too round to be a space-efficient cup:



5. In **Edit** mode, edges can be marked with a "**crease**" to indicate to the **Subsurf** modifier that they should be sharpened.
6. With no points selected (*A*), in edge or point select mode (*Ctrl + Tab*), hold down *Alt* on the keyboard and select (click on) one of the edges around the top lip of the cup to select all the points in a loop around the top of the mug:

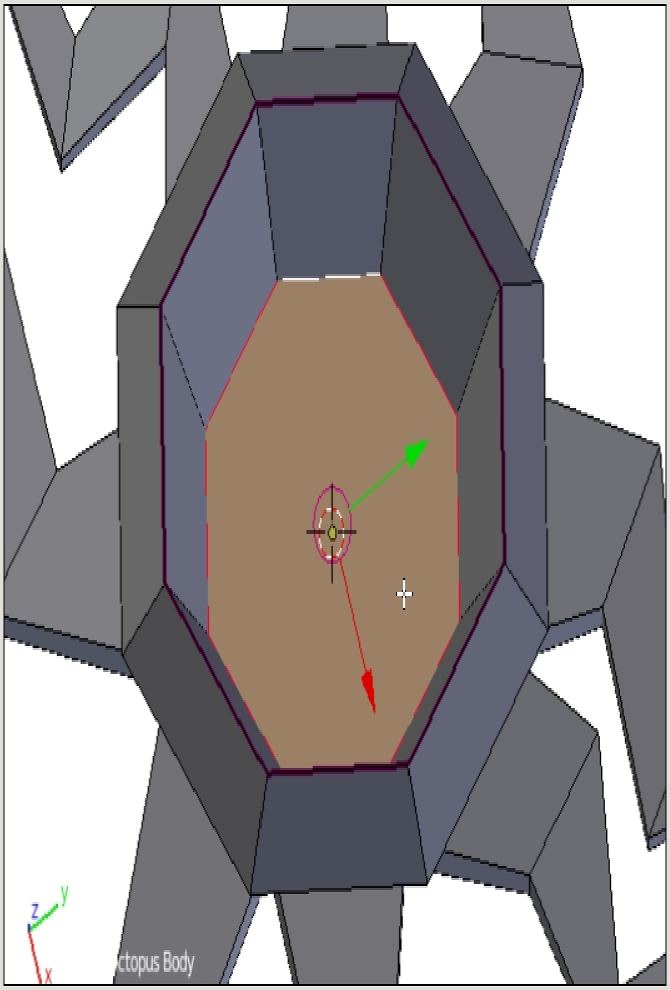


7. From the **3D View** menu, go to **Mesh** | **Edges** | **Edge Crease** or press *Ctrl* + *E* on the keyboard. Move the mouse pointer up or down to increase or decrease the amount of creasing applied to the edge, until it looks good:

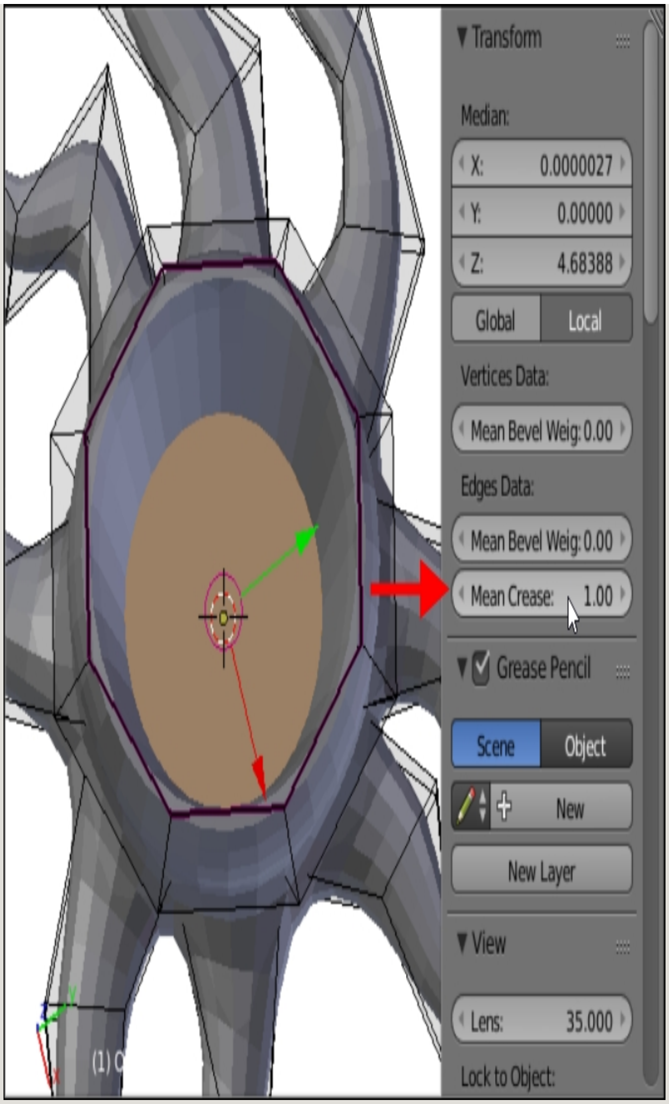


8. Hold *Alt* and select (click on) one of the edges at the bottom of the cup to loop select all the points around the

bottom of the cup. It may help to switch to **Wireframe** mode (*Z*) or temporarily turn off the **Subsurf** modifier:



9. This time, set the crease value by finding the **Mean Crease** setting in the **Properties** tab (*N*). This setting can be any decimal number from `0.0` (off) and `1.0` (maximum). Click on the setting and enter `1`, and then press *Enter* to set the value. Now the bottom of the cup is flat:

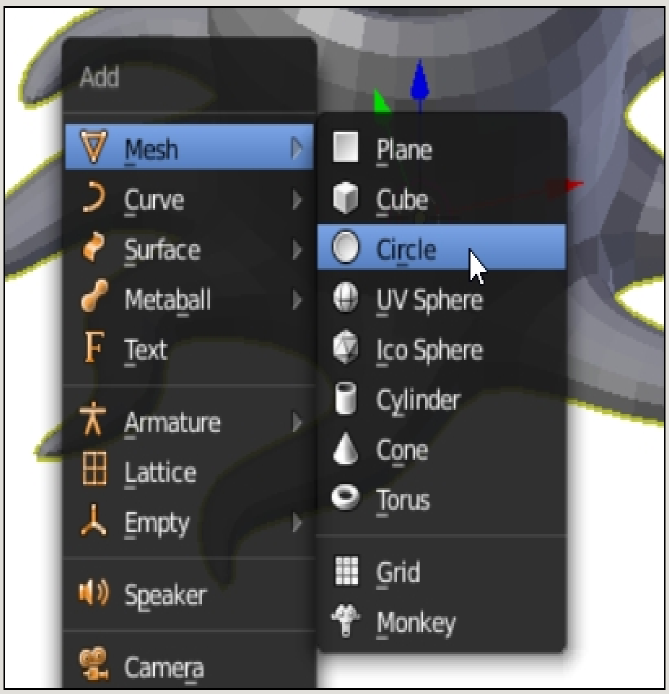


This sort of edge creasing is a powerful way to control the **Subsurf** modifier. Notice that the bottom of the cup is still circular even though the original mesh is an octagon. **Subsurf** still smooths out the other edges even if it's told to crease some—it's very smart!

Adding a face

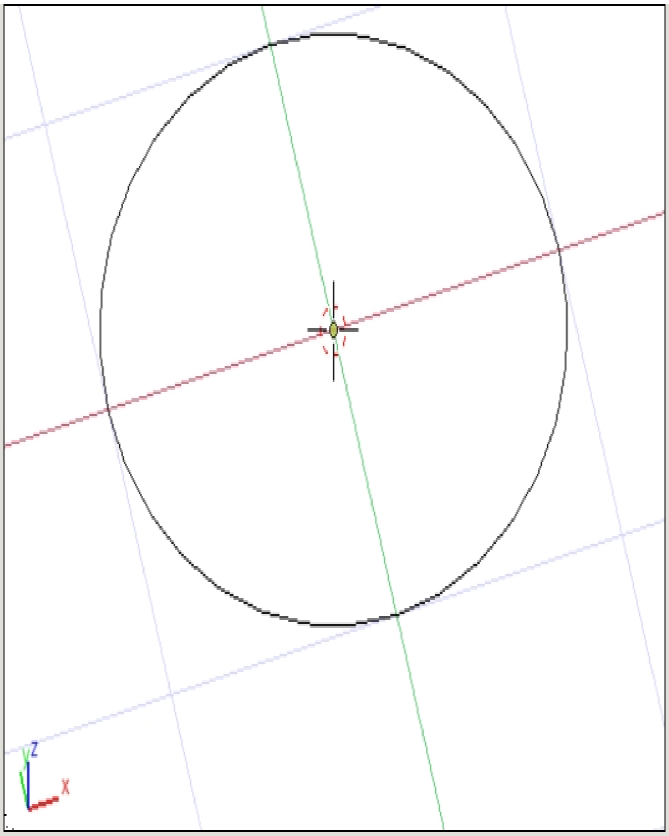
Functionally, the design is complete, but as the old saying goes, it's all in the details. This little octopus would be much cuter with a face. Here's how you give it one:

1. Make sure **Edit** mode is off (*Tab*) and that the 3D cursor is at the origin (*Shift* + *C*).
2. Create a circle (*Shift* + *A*) by navigating to **Mesh** | **Circle**:

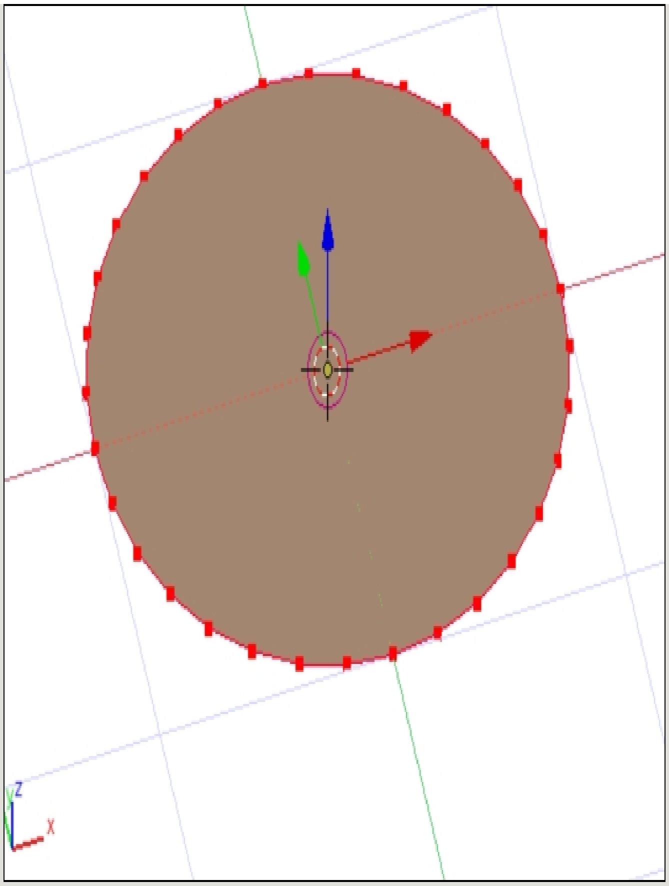


3. Again, the newly created circle is hidden inside the octopus, so with the circle still selected, switch to **Local** view (Numpad */*).

   Since this circle will become the face, it's good to name it Face now, using the same steps for naming objects from before.
4. A circle in Blender is a flat object that is just a ring of points with no face. But that's easy enough to fix:

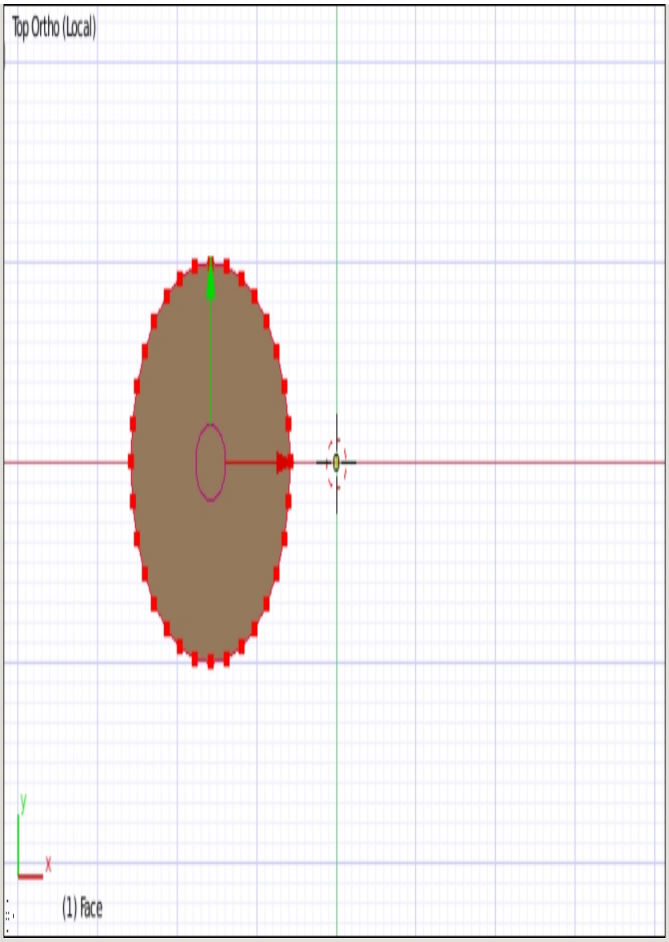


5. Switch to **Edit** mode (*Tab*).
6. With all the points or edges selected (*A*) from the **3D View** menu, navigate to **Mesh** | **Faces** | **Make Edge/Face**, or press *F* on the keyboard:



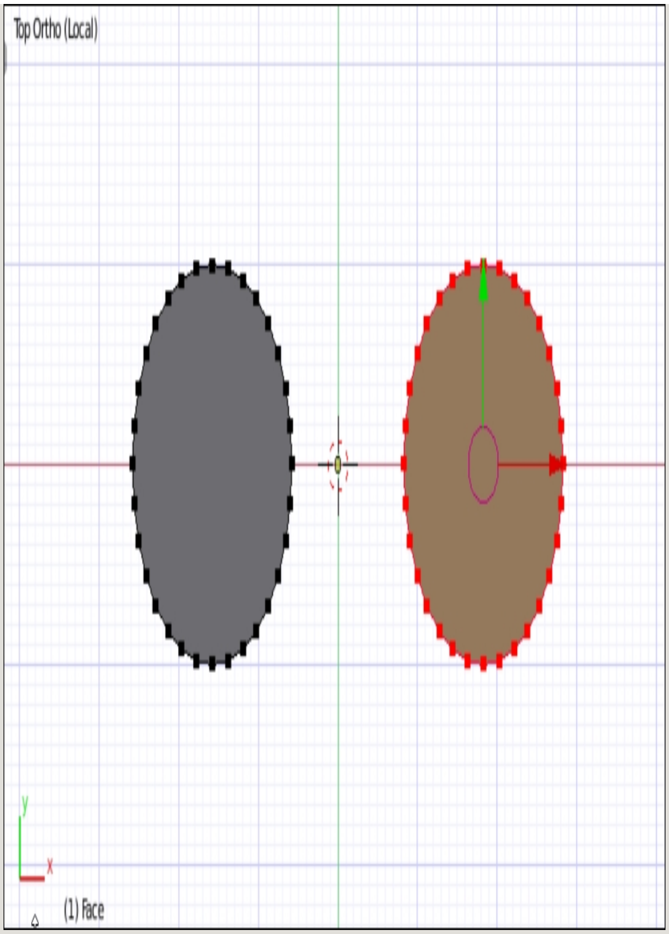
The Make Edge/Face command attempts to make a connected face from the selected vertices or lines. It can be quite intelligent, but if the points are not all flat, the edges may twist unexpectedly.

7. While still in **Edit** mode, switch to top view (Numpad *7*), and with all the points still selected, move (*G*) them to the left of the green Y-axis line:



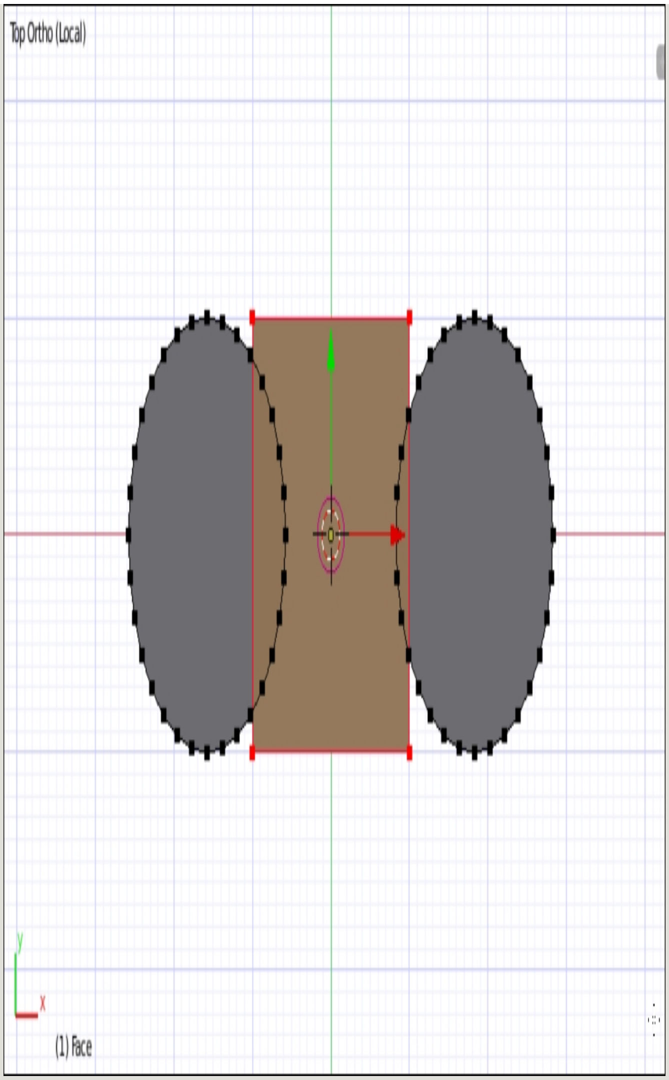
Duplicating in **Edit** mode doesn't create a new object. Whatever is selected is duplicated, but it remains a part of the object it was duplicated from. Points, lines, and faces can all be duplicated and transformed within the same object, even if it looks like they're separate parts. In this way, a face with separate eyes and mouth can be created that will still all be one face object.

8. Duplicate (*Shift* + *D*) all the points and place the duplicated points on the opposite side of the green Y axis line:

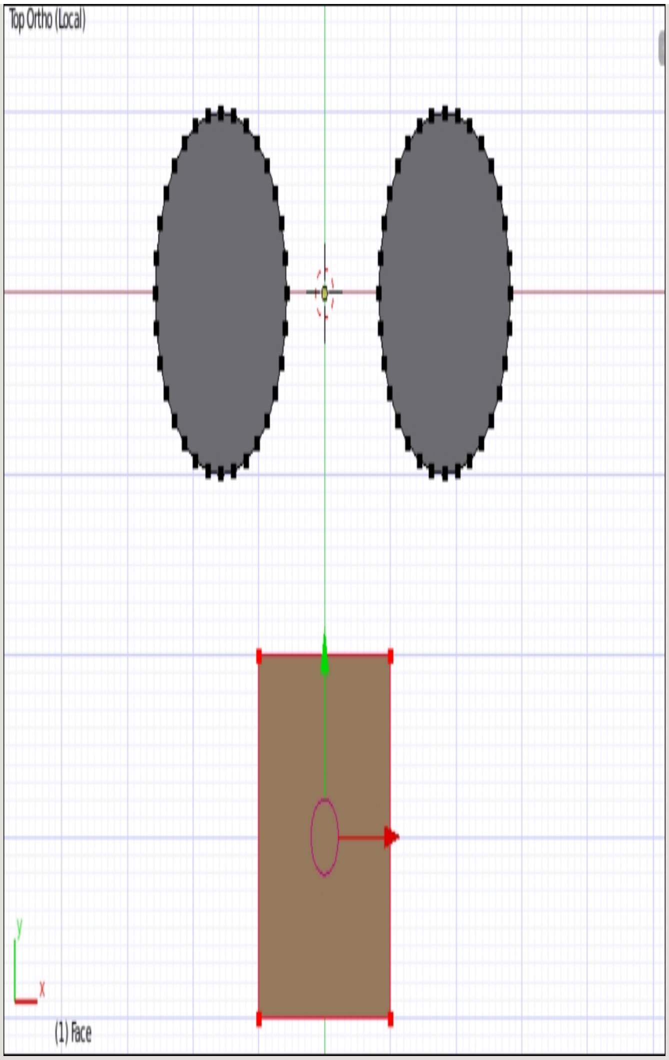


Both eyes are still a part of the face object, even though they're separate. In the same way, objects created while in **Edit** mode are likewise added to the existing object.

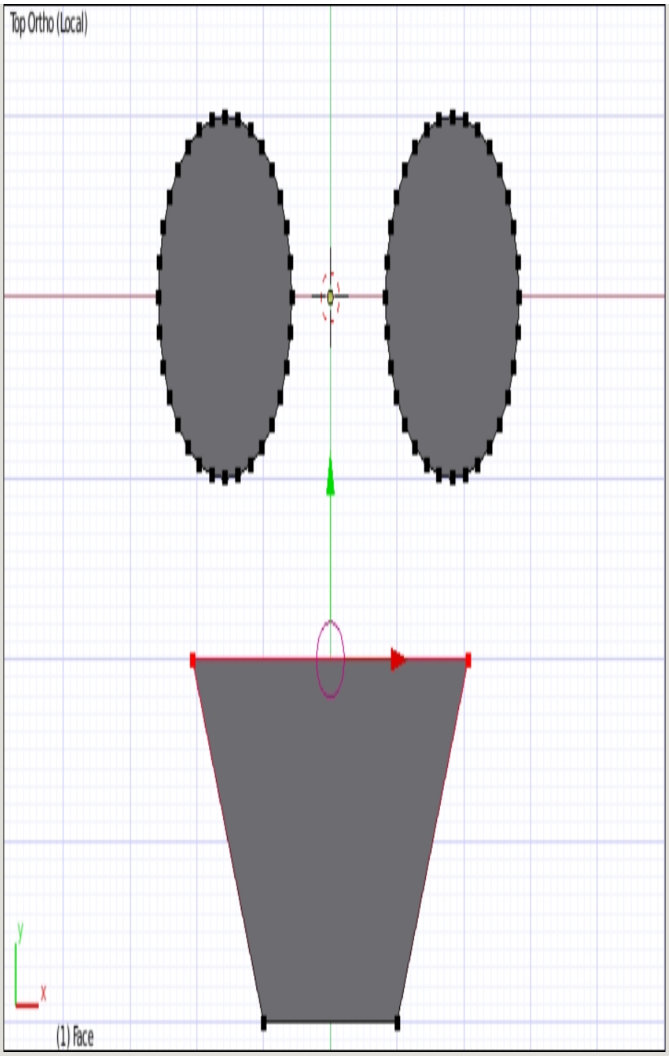
9. While still in **Edit** mode (making sure the 3D cursor is still at the origin), add (*Shift* + *A*) a plane to the face object by navigating to **Mesh** | **Plane**:



10. Move (*G*) the plane down along the Y-axis (*Y*) (about 3 units):

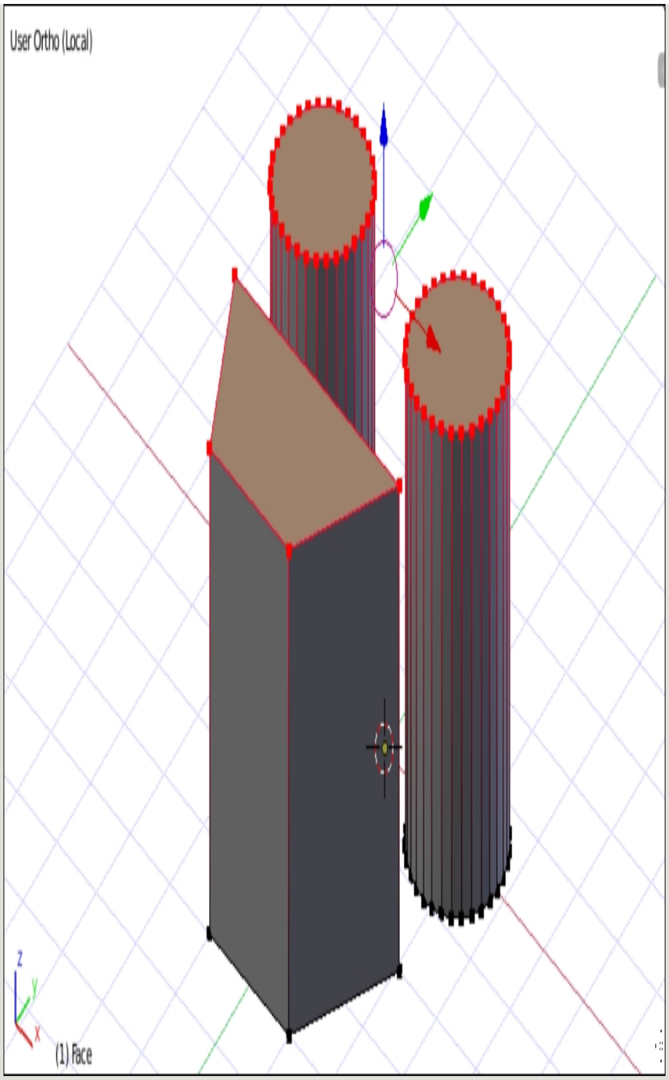


11. Deselect all the points (*A*).
12. Select the two points at the top of the plane.
13. Scale (*S*) them up about twice to make the square a smiling mouth:

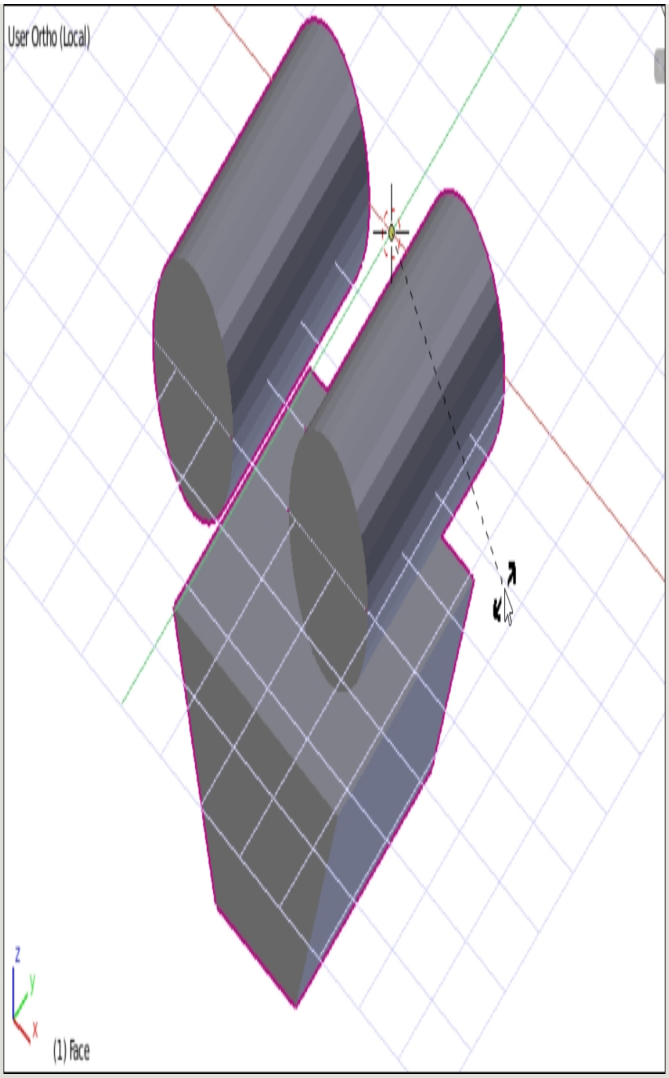


The shapes now look like a face, but only in two dimensions. If seen from a different angle, it's clear that the face is very flat. To give it the additional dimensions necessary, follow these steps:

14. Select all the points (*A* twice) and extrude them (*E*) about 4 or 5 units:

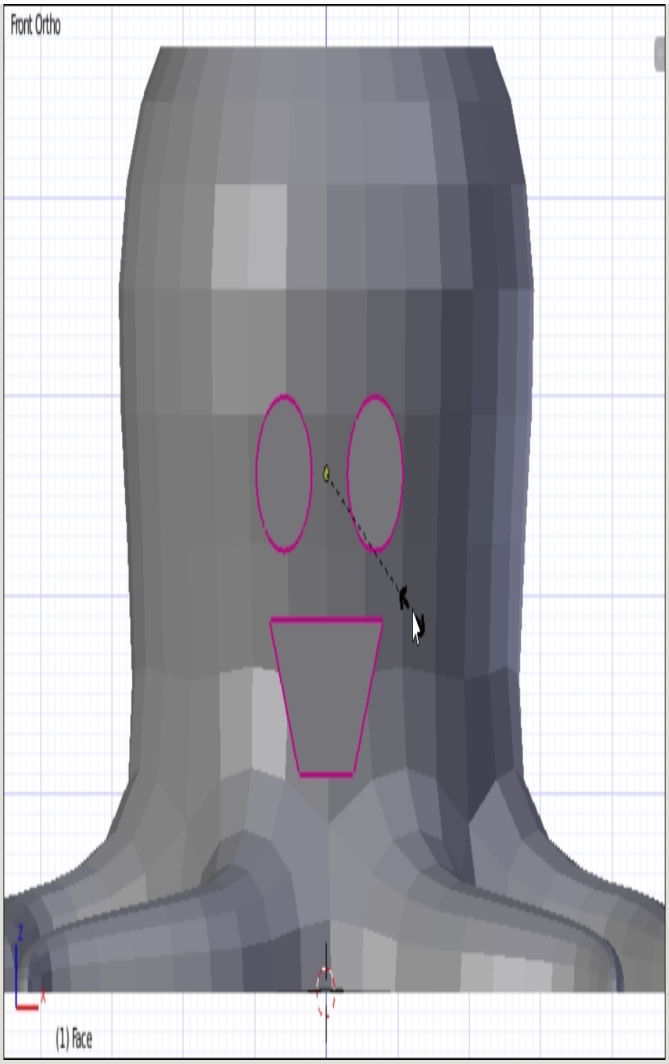


15. Exit **Edit** mode (*Tab*). The face template is now complete and just needs to be scaled and positioned.
16. Rotate (*R*) the face object around the `X` axis (*X*) `90` degrees (*90*):

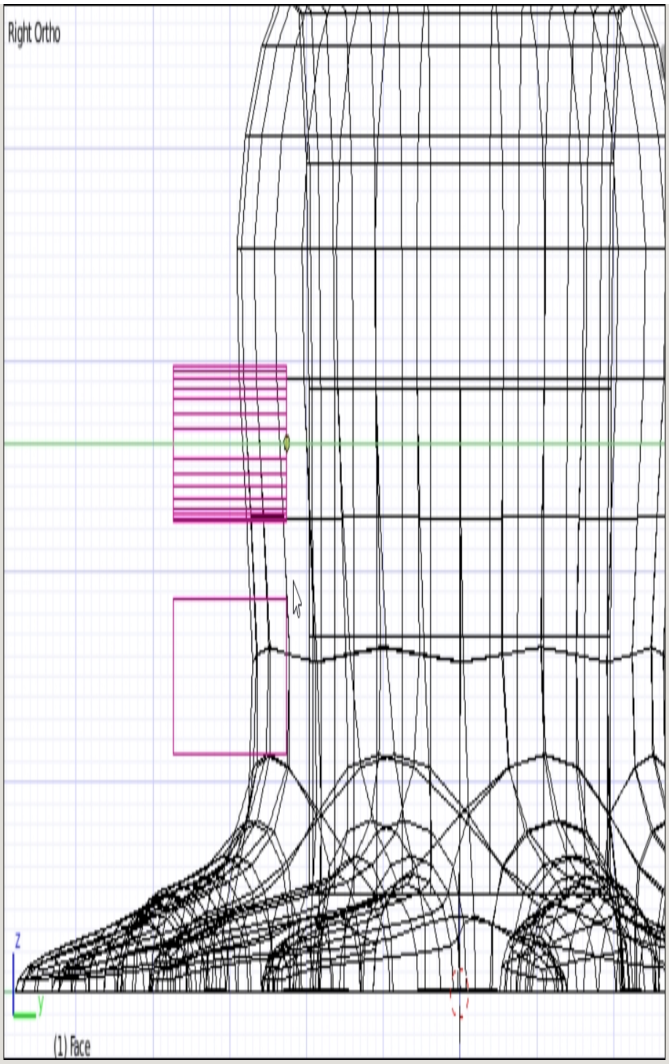


17. Exit **Local** view (Numpad /). The face is hidden inside the body, so move (*G*) it along the Z and Y axes (*Shift* + *X*) until it's outside the octopus body:

18. In front **Ortho** view (Numpad *1*), scale the face (*S*) and move (*G*) it along the Z axis (*Z*) until it's positioned properly:

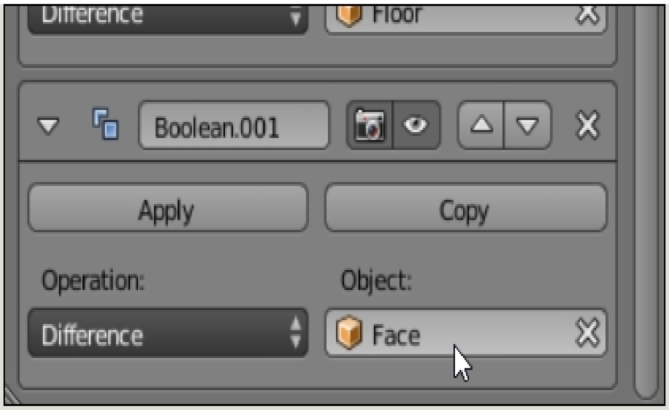


19. In the side **Ortho** view (Numpad *3*), move (*G*) the face along the Y axis (*Y*) until it's partway inside the body:

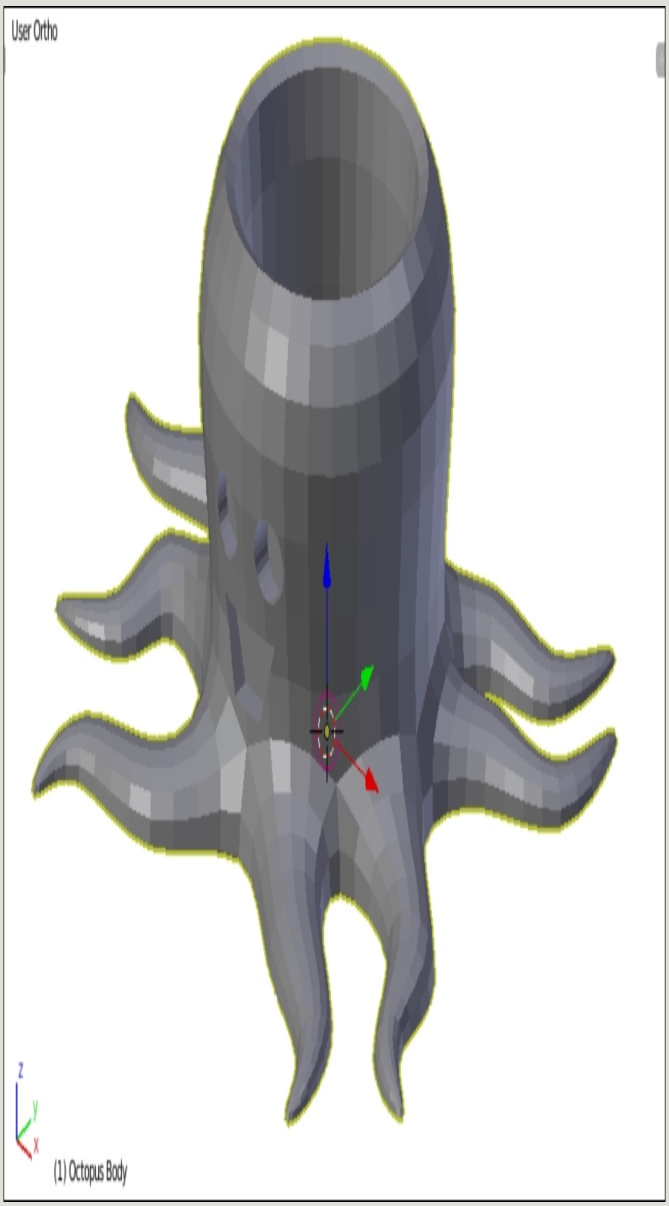


It may help to use **Wireframe** view to ensure the face is deep enough but not too deep. About four or five small squares' depth (or `0.4` or `0.5` real-world millimeters) is perfect. If it's too deep, it may cut all the way through into the cup space and create an overhang problem. The top of the mouth will rely on some bridging during prints, and that's okay.

20. Now, select the body and in the **Modifier** tab, add a **Boolean** modifier. Change the settings of the **Boolean** modifier to be a **Difference** operation with the face object:

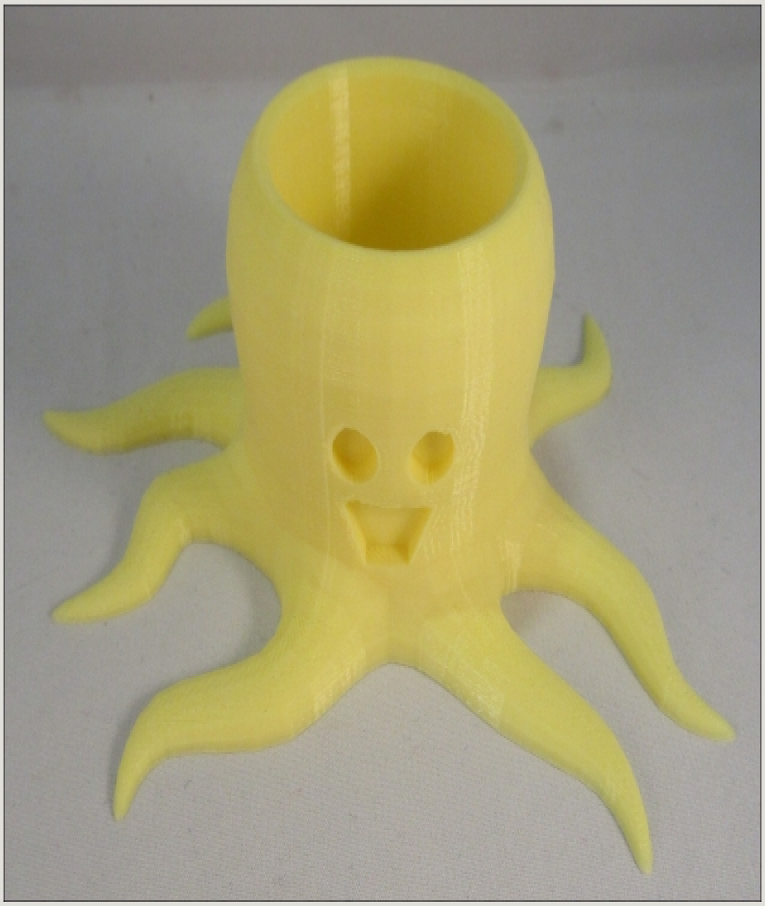


21. The effect of this new modifier won't be visible immediately, so select the face object and hide it (*H*):



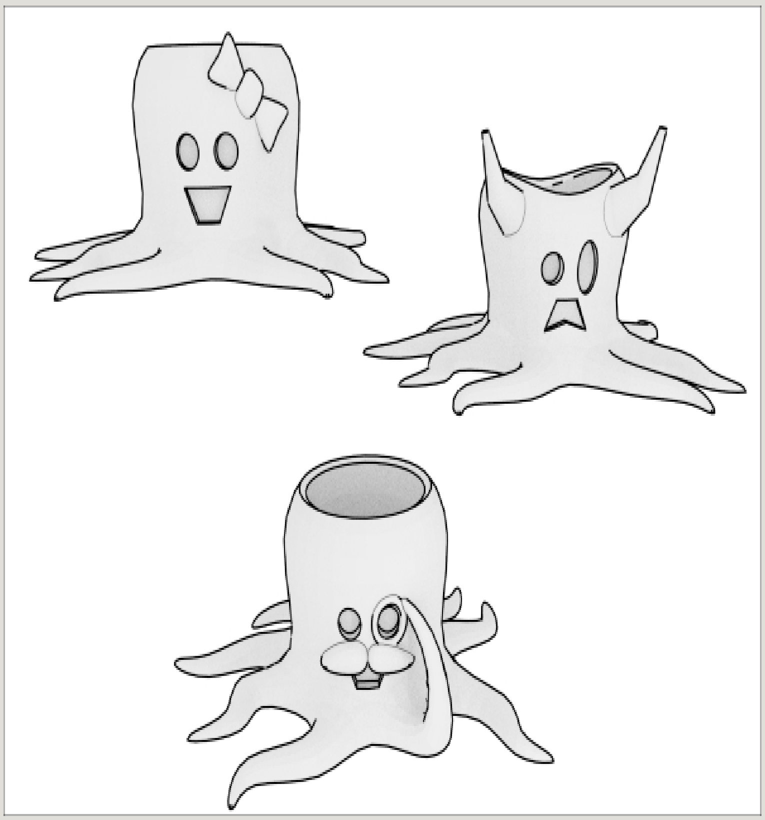
Your octopus pencil holder is now complete.

Select the body, and in the **Info Panel** menu (at the top of the screen), go to **File** | **Export** | **Stl (.stl)** to export the 3D print-ready octopus holder. Be sure to choose a directory where you can find the file later in order to send it to the printer.



Finishing touches

The octopus is done now, but don't stop there. Use your imagination to make this design your own. Add accessories and personality, change the base shape, and come up with your own design to make this something you'll want to share. Just remember the rules about overhangs in your design, and the sky is the limit:



Summary

This project was functional but also cute. The techniques for vertex editing basic shapes in **Edit** mode with extrusion and loop cuts and then using the mesh smooth and **Boolean** modifiers to further modify them can be used to create an endless number of projects of any type and shape. Not all designs need to be cute, but they also don't all need to be purely practical. Combine a practical design with an aesthetic element to add personality and really take advantage of what 3D printing can offer.

Happy designing!

Index

O

P

S

T

V